TEXT FEATURE SELECTION FOR RELEVANCE DISCOVERY: A FUSION-BASED APPROACH

Abdullah Samaran A AL HARBI

M.IT(SE), GC(CS&WS), GC(CN), GC(EC), B.Sc-Hons(CS)

Submitted in fulfilment of the requirements for the degree of
Doctor of Philosophy

School of Computer Science
Science and Engineering Faculty
Queensland University of Technology

2020

Copyright in Relation to This Thesis

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Signature: QUT Verified Signature

Date: 10/04/2020

To the beautiful soul I recently lost.

To the loss that made me despair in grief.

To my beloved late father.

To him, I sincerely dedicate this thesis.

May Allah bestow his mercy on him.

Amen.

Abstract

This thesis presents innovative and significantly effective fusion-based TFS models and frameworks to overcome the above problems in LDA and the relevant features discovered by existing relevance discovery models. The proposed models—SIF, SIF2 and UR—extend multiple random sets to model and, therefore, understand the complex relationships between different entities that affect the weighting process of topical terms in a collection of documents. The models effectively fuse different features to (1) generalise terms' weights to the collection level, (2) alleviate the impact of local terms' frequency, (3) estimate document segment relevancy and (4) relax the assumption of a globally generalised topical term weight. This thesis also proposes two TFS frameworks, USIF and SSIF, which adopt the idea of differentiating between feature selection and feature weighting processes at two stages to discover relevant features. USIF integrates topic modelling, document clustering and global statistics to reduce uncertainties and the impact of highly frequent topics or sub-topics in an unsupervised context. Conversely, SSIF is supervised and incorporates support vector machines, topic modelling and collection statistics to reduce the impact of terms that frequently appear in both positive and negative topics in a document collection, as well as the uncertainties available in relevant documents. All the proposed models and frameworks are extensively evaluated for information filtering using a series of experiments based on 50 collections from the standard RCV1 dataset and their TREC assessors' relevance judgements. The ability of the proposed models and frameworks in ranking relevant terms in these collections is also tested. The experimental results, measured by seven different performance metrics, the percentage of change and the Student's t-test, show that SIF, SIF2, UR, USIF and SSIF significantly outperform all state-of-the-art and popular baseline TFS models, regardless of the type of text feature they adopt, the fusion strategies they adhere to or the mining and learning algorithm they use.

Keywords

Text Feature Selection

Relevance Discovery

User Information Needs

Random Sets

Extended Random Sets

Uncertainty Reduction

Term Weighting

Feature Re-Ranking

Weight Scaling

Global Statistics

Text Feature Fusion

Early Fusion

Late Fusion

Hybrid Fusion

Passage Relevance

Topic Modelling

Latent Dirichlet Allocation

Text Mining

Information Filtering

Ranking Relevant Terms

Acknowledgments

All praise and thanks be to the Almighty Allah (SWT) for all his grace and blessings. My PhD journey would not have been fruitful without his mercy and generosity.

This research project would never have been successful without the help and support of many people. First, I would like to express my sincere gratitude to my principal supervisor, Professor Yuefeng Li, for his constant help and support. His patience, encouragement, guidance, knowledge in the field and wisdom in research were more than I needed to succeed in my PhD studies. I cannot thank you enough for believing in me throughout my candidature.

Many thanks and much appreciation also goes to my associate supervisor, Associate Professor Yue Xu, for her invaluable guidance and comments on my research work. I would also like to thank the external examiners for their suggestions and constructive comments. I also thank all my friends and colleagues in the data science research group, particularly my AI-based data analysis lab members: Dr Mubarak Albathan, Dr Md Abul Bashar, Dr Yutong Wu and Dr Khaled Albishre.

I would also like to thank the School of Computer Science and the Science and Engineering Faculty at the Queensland University of Technology (QUT) for providing me with everything I needed to conduct my research and financial support for my travels. As I have studied for my English course, graduate certificates, master's degree and most recently PhD all at QUT, I find myself indebted to the general public of QUT and Australia for the amazing environment and friendly atmosphere.

Also, I would like to thank and acknowledge the editors at Capstone Editing®, who provided copyediting and proofreading services, according to the guidelines laid out in the university-endorsed national 'Guidelines for Editing Research Theses'.

A special *thank you* goes to my mother for all her prayers and encouragement. Many thanks also to my brothers and sisters for their emotional support. Words alone will never be enough to thank my wife W. Alluhaybi, my son Mohammad and my daughter Lana for being helpful, supportive and, ultimately, very patient during my busy times. Also, I cannot forget to thank my new family member, Leena, for waking me up so early crying for some milk!

Finally, I am grateful to my employer, Umm Al-Qura University, for generously providing me with a full scholarship, without which I could not have pursued my postgraduate studies overseas. Also, the people at the Saudi Arabian Cultural Mission in Canberra deserve much appreciation and gratitude for all their help and support to all Saudi scholars in Australia, including myself.

List of Publications

1. Abdullah Alharbi, Yuefeng Li and Yue Xu, ‘Enhancing topical word semantic for relevance feature selection’, in *Proc. IJCAI Workshop on Semantic Machine Learning*, Melbourne, (vol. 1986), 2017, pp. 27–33.

2. Abdullah Alharbi, Yuefeng Li and Yue Xu, ‘Integrating LDA with clustering technique for relevance feature selection’, in Peng W., Alahakoon D., Li X. (eds) *AI 2017: Advances in artificial intelligence*. Lecture notes in computer science. Cham: Springer, 2017, vol. 10400, pp. 274–286. (Awarded Best Student Paper Award).

3. Abdullah Alharbi, Yuefeng Li and Yue Xu, ‘Topical term weighting based on extended random sets for relevance feature selection’, in *Proc. Intern. Conf. on Web Intelligence*, Leipzig, Germany, 2017, pp. 654–661. (Awarded Best Paper Award).

4. Abdullah Alharbi, Yuefeng Li and Yue Xu, ‘An extended random-sets model for fusion-based text feature selection’, in Phung D., Tseng V., Webb G., Ho B., Ganji M., Rashidi L. (eds) *Advances in knowledge discovery and data mining. Lecture notes in computer science*. Cham: Springer, 2018, vol 10939, pp. 126–138. (**Selected for long presentation [10% accept rate]**)

5. Abdullah Alharbi, Md Abul Bashar and Yuefeng Li, ‘Random-sets for dealing with uncertainties in relevance feature’, in Li X., Mitrovic T., and Xue B. (eds) *AI 2018: Advances in artificial intelligence. Lecture notes in computer science*. Cham: Springer, 2018, vol 11320, pp. 656–668. (**Selected for long presentation**).

Table of Contents

Nomenclature

Abbreviations

AI	Artificial Intelligence
BKM	Bisecting K-Means
BoW	Bag-of-Words
CF	Cluster Frequency
CFS	Correlation-based Feature Selection
CHINC	Constrained Heterogeneous Information Network Clustering
CTF	Conceptual Term Frequency
DF	Document Frequency
DM	Data Mining
EM	Expectation–Maximization
ERS	Extended Random Set
FCBF	Fast Correlation Based Filter
FCP	Frequent and Closed Pattern
FP-tree	Frequent Pattern-tree
FS	Feature Selection
GSDMM	Gibbs Sampling for Dirichlet Multinomial Mixture
hPAM	Hierarchical Pachinko Allocation Model
IAP	Interpolated Average Precision
ID3	Iterative Dichotomiser 3
IDF	Inverse Document Frequency

IF	Information Filtering
IG	Information Gain
IR	Information Retrieval
KDD	Knowledge Discovery in Databases
KDT	Knowledge discovery in Textual Databases
kNN	k-Nearest Neighbours
LASSO	Least Absolute Shrinkage and Selection Operator
LCSH	Library of Congress Subject Headings
LDA	Latent Dirichlet Allocation
LSA/LSI	Latent Semantic Analysis/Indexing
MAP	Mean Average Precision
MedLDA	Maximum entropy discrimination Latent Dirichlet Allocation
MI	Mutual Information
ML	Machine Learning
MP	Master Pattern
MPBTM	Maximum matched Pattern-Based Topic Model
mRMR	Minimum Redundancy Maximum Relevance
nDCG	Normalized Discounted Cumulative Gain
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
Okapi BM25/BM25	Best Matching 25
PAKDD	Pacific-Asia Conference on Knowledge Discovery and Data Mining
PAM	Pachinko Allocation Model
PBTM-FCP	Frequent Closed Pattern-Based Topic Model
PBTM-FP	Frequent Pattern-Based Topic Model
PCM	Pattern Co-occurrence Matrix
PDM	Pattern Deploying Model

PDS	Pattern Deploying based on Support/System
PF	Paragraph Frequency
PLSA/PLSI	Probabilistic Latent Semantic Analysis/Indexing
POS	Part-Of-Speech
PTM	Pattern Taxonomy Model
RCV1	Reuters Corpus Volume 1
RFD	Relevant/Relevance Feature Discovery
RRT	Ranking Relevant Terms
RS	Random Set
SCSP	Specific Closed Sequential Pattern
SF	Sentence Frequency
SIF	Selection of Informative Feature
SMART	System for the Mechanical Analysis and Retrieval of Text
SML	Semantic Machine Learning
SP	Sequential Pattern
SPADE	Sequential PAttern Discovery using Equivalence classes
SPBTM	Significant matched Pattern-Based Topic Model
SPEC	Spectral Feature Selection
SSIF	Supervised Selection of Informative Feature
STM	Structural Topic Model
SVM	Support Vector Machine
TF	Term Frequency
TFIDF	Term Frequency Inverse Document Frequency
TFS	Text Feature Selection
TM	Text Mining
TNG	Topical N-Gram
TREC	Text REtrieval Conference

UR	Uncertainty Reduction
USIF	Unsupervised Selection of Informative Feature
VSM	Vector Space Model
WI	Web Intelligence

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Text documents grow exponentially and constitute more than 80% of the unstructured data available on the web or in private storage [Dhar, 2013, Khan et al., 2010]. Some forms of unstructured text include emails, tweets, reports, articles, logs and reviews [Blei, 2012, Dhar, 2013]. These documents contain invaluable information that needs to be automatically extracted for the success of many organisations and businesses [Bashar et al., 2014, Dhar, 2013]. However, it is particularly challenging for traditional text mining (TM) and machine-learning techniques to find useful information in textual data due to the size and nature of text in which synonymy, polysemy and noise are commonly inherited problems [Croft, 2000, Li et al., 2015, 2010, 2012]. As a dimensionality reduction technique, feature selection plays a major role in knowledge discovery in databases by improving accuracy and reducing the complexity of many data-mining and machine-learning algorithms [Aphinyanaphongs et al., 2014, Cai et al., 2010, Dasgupta et al., 2007]. This can be done automatically by selecting a subset of relevant features and removing irrelevant, redundant and noisy features [Albathan et al., 2014, Li et al., 2015, Zhong et al., 2012].

1.1.1 Text Feature Selection

Relevance discovery models endeavour to mine, interpret, understand and rank relevant features that specifically represent what the user needs [Gao et al., 2014b, Li et al., 2015, 2010, Man et al., 2009, Zhong et al., 2012]. User information needs can be explicitly expressed by a search query or inferred from the user's interests or search profile, in the form of a set of documents [Algarni et al., 2010, Li and Yao, 2002a, Tao et al., 2011, Yuefeng and Ning, 2006]. Discovering and selecting features that are relevant to the user's needs or interests is challenging, and remains the subject of much research [Gao et al., 2015, Li et al., 2015, 2010]. In the absence of a user query, relevance feedback—as a set of documents that can be relevant or irrelevant to a particular topic—provides an effective way to identify text features that can be used to describe user information needs [Algarni et al., 2010, Tao et al., 2011]. However, guaranteeing the quality of these features is challenging, as text documents tend to have many uncertainties in addition to a large number of terms, patterns, noise and multiple unbalanced sub-topics [Alharbi et al., 2018a, 2017b, Gao et al., 2015, Li et al., 2015]. These problems have interested TM, natural language processing (NLP), machine learning (ML), information filtering (IF) and information retrieval (IR) research communities from both theoretical and empirical perspectives [Li et al., 2015, 2010].

In recent decades, a large number of text feature selection (TFS) techniques have been developed by these research communities. Within each community, TFS models and frameworks have been categorised based on first, the intrinsic details of the selection algorithm, into filter, wrapper, embedded and hybrid [Bolón-Canedo et al., 2013, Li et al., 2017a, Liu and Yu, 2005]; second, whether it requires labelled training data, into supervised, semi-supervised or unsupervised methods [Li et al., 2017b, Wang et al., 2017, Zhao et al., 2013]; and third, the structure of text feature being used, into simple, such as the term-based methods, or complex, like the phrase-based, pattern-based, topic-based, concept-based or hybrid techniques [Li et al., 2015, 2010]. In this thesis, a fourth categorisation for TFS techniques is adopted based on integrating data fusion strategies like early, late and hybrid approaches [Kozorovitsky and Kurland, 2011b, Lillis et al., 2006, Wu et al., 2014] with the simple or complex structure of text (identified above as the third classification). Thus, a TFS model is considered an early fusion model if it uses simple, low-level terms (i.e., no semantic information is considered between the terms) [Alharbi et al., 2018b, Balazs and Velásquez, 2016] and a late fusion model when complex, high-level and semantically rich features (e.g., phrases, patterns, concepts, topics or a combination of these text features) are used [Alharbi et al., 2018b]. Also, a hybrid fusion model can be developed by integrating different early and late fusion models for even better performance [Atrey et al., 2010, Baltrušaitis et al., 2019].

A TFS model for relevance discovery selects the most informative text features, such as terms [Combarro et al., 2005, Man et al., 2009, Zheng et al., 2004], phrases [Fürnkranz, 1998, Sebastiani, 2002, Shirakawa et al., 2015], patterns [Algarni and Li, 2013, Li et al., 2015, Yuefeng and Ning, 2006], concepts [Li and Zhong, 2004, Shehata et al., 2010, Tao et al., 2011], topics [Blei et al., 2003, Deerwester et al., 1990, Hofmann, 2001] or different combinations of these features [Gao et al., 2015, Li et al., 2015, Wang et al., 2007] that describe user information needs. The selected features are used to represent documents to help TM algorithms, such as filtering [Gao et al., 2015, Li et al., 2011, 2012], classification [Li et al., 2017c, Shehata et al., 2007, Yang and Pedersen, 1997] and clustering [Cai et al., 2010, Liu et al., 2003, Shehata et al., 2010], to be: (1) effective by increasing their accuracy, (2) efficient by reducing the dimensionality of the feature space and thus, the algorithms take less computational time and (3) tractable and understandable [Liu et al., 2005, Song et al., 2013]. By focusing on the selected features, it is possible to understand how and why such algorithms behave a certain way and produce certain results [Bashar and Li, 2017, 2018, Bashar et al., 2017].

While each text feature has strengths and weaknesses, latent topics that are extracted by topic modelling algorithms have received much attention in many applications [Blei et al., 2010a, Blei, 2012, Blei et al., 2003, Hofmann, 2001]. However, exploiting these topics for TFS for relevance discovery is still an open research problem, as these algorithms did not show encouraging results in many recent studies [Alharbi et al., 2017c, Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2017, 2015]. Therefore, the focus of this thesis is developing fusion-based TFS models and frameworks for relevance discovery. These models and frameworks integrate early and late fusion strategies with other learning algorithms to fuse different features to manage uncertainties in relevance feedback and overcome the limitations of topic modelling algorithms and other TFS techniques in selecting informative topical terms that describe user information preferences.

1.1.2 Text Feature Fusion Strategies

As there is no single text feature that can encompass all evidence of relevance available in a set of documents that discusses user information needs, text feature fusion offers an approach that integrates different features of text with various degrees of relevance for better performance in many IR, IF, TM and ML applications [Anava et al., 2016, Li et al., 2013, Pickens and Golovchinsky, 2008]. The fused features are more reliable and can be used to model uncertainty and thus, increase confidence and robustness of learning algorithms [Balazs and Velásquez, 2016, Croft, 2000, Esteban et al., 2005]. Two main fusion approaches have been employed in IR and ML, commonly known as the early and late fusion strategies [Balazs and Velásquez, 2016, Zhang and Balog, 2017]; however, there is also a hybrid strategy that combines the first two approaches [Atrey et al., 2010, Baltrušaitis et al., 2019, Snoek et al., 2005].

1.1.2.1 Early Fusion Strategy

Term-based Models

Most existing TFS models for relevance discovery adopt the early fusion strategy in which no semantic information is considered among the fused features [Alharbi et al., 2018b]. Popular examples are the term-based methods, such as term frequency-inverse document frequency (TFIDF) [Salton and Buckley, 1988], mutual information (MI) [Manning et al., 2008b], information gain (IG) [Yang and Pedersen, 1997], Gini-index (GI) [Zhu and Lin, 2013], Chi-Square (χ^2) [Chen and Chen, 2011], best matching 25 (BM25) [Robertson and Zaragoza, 2009], Rocchio algorithm [Rocchio, 1971], least absolute shrinkage and selection operator (LASSO) [Tibshirani, 1996], ranking support vector machine (SVM) [Joachims, 2002] and many others. These models are efficient and have been developed based on sophisticated mathematical and statistical weighting theories [Li et al., 2015, 2010]. However, their use of low-level terms (i.e., individual words) makes them sensitive to noise and semantic-related issues, such as synonymy and polysemy problems [Li et al., 2015, 2012, Zhong et al., 2012]. A synonymous word is lexically different from another semantically identical word (e.g., ‘man’ and ‘guy’) [Wu, 2007]; conversely, a polysemous word is lexically identical but has different contextual meanings (e.g., ‘newspaper’ as a company and as a physical item) Sebastiani [2002].

In IR, term-based models suffer from mismatching and overloading as a result of synonymy and polysemy issues [Liu et al., 2016, Yuefeng and Ning, 2006]. Mismatching occurs when query terms do not exhaustively represent the user’s search topic [Tao et al., 2011, Yuefeng and Ning, 2006]. For example, documents that only discuss the subject of ‘knowledge discovery’ will be missed if the user uses the query ‘data mining’, knowing that data mining and knowledge discovery are closely related subjects. This problem is usually referred to as the synonymy problem (synonymy can cause information mismatch) [Bashar and Li, 2018, Li et al., 2012]. Conversely, information overload can be caused by the polysemy problem when query terms can be used in different contexts [Bashar et al., 2016, Tao et al., 2011]. A common example is the use of the query term ‘Java’, whether it means coffee, the island of Java in Indonesia or the Java programming language [Bashar et al., 2016, 2017, Bing et al., 2015]. Further, by using the bag-of-words (BoW) representation, term-based methods ignore word order in documents and consequently miss the semantic relationships between these words [Li et al., 2009a, Turney and Pantel, 2010].

1.1.2.2 Late Fusion Strategy

Phrase-based Models

TFS techniques that adopt the late fusion approach utilise different high-level features that have semantic information, such as phrases, patterns, concepts, topics or a combination of these features [Zhou et al., 2010]. Phrase-based TFS models use phrases (e.g., n-grams) because they are (1) more discriminative, (2) better able to contain semantic information than single words and (3) less ambiguous [Li et al., 2015, 2011, 2010, 2012]. The n-grams are commonly extracted using different values for 'n', which are experimentally specified [Albathan et al., 2013, Fürnkranz, 1998, Wang et al., 2012]. Language models (e.g., n-grams models) [Lavrenko and Croft, 2001, Robertson and Zaragoza, 2009, Wang et al., 2007] are particularly relevant in the field where phrases, as a sequence of words, are probabilistically formulated and these lexical features extracted using the n-grams (e.g., unigram (n=1), bi-gram (n=2), tri-gram (n=3), etc.) to maintain the terms' dependencies. However, published phrase-based experiments do not show encouraging results compared to term-based ones [Gao et al., 2015, Li et al., 2015, Moschitti and Basili, 2004, Scott and Matwin, 1999, Wu et al., 2006]. Wu et al. [Wu et al., 2006] argue that this is because phrases' statistical attributes are inferior to terms, they suffer from redundancy and noise, and meaningful phrases suffer from the low-frequency problem that makes them hard to distinguish and thus, select.

Whether heuristically or probabilistically justified, phrase-weighting functions assign scores to phrases to represent their relevance to user information needs. However, they treat a phrase as an atomic unit of meaning and assume its terms are equally important to the user's needs [Shirakawa et al., 2015, Wang et al., 2007]. This assumption can be too simple for discovering relevant features. For example, a traditional n-gram model would uniformly weigh the phrase 'President Bill Clinton' even though each term in the phrase has a specific meaning and can be more important than the others and thus, should be assigned different representative weights [Hammache et al., 2014, Metzler and Croft, 2005, Shi and Nie, 2009]. It can also be too simple to assume that phrases are more representative because they carry semantic meaning and less ambiguity than terms [He et al., 2011, Lv and Zhai, 2009, Miao et al., 2012]. For example, if a short document contains 'deep hierarchical reinforcement learning', then a popular phrase like 'deep learning' cannot be used as a representative feature for the document because 'deep' and 'learning' do not sequentially appear in the document, even though the phrase is relevant to the document [Gao, 2015].

Pattern-based Models

To overcome the shortcomings of term- and phrase-based methods, different pattern-based techniques have been introduced [Algarni et al., 2010, Li et al., 2015, 2011, 2010, 2012, Wu et al., 2006, 2004, Zhong et al., 2012]. A pattern, as a set of associated terms, has been used for feature selection [Albathan et al., 2013, 2014, Li et al., 2015, 2010, Zhou et al., 2011]. Many efficient pattern-mining algorithms have been developed in data mining, such as Apriori-like algorithms [Agrawal and Srikant, 1994], Pre-fixSpan [Pei et al., 2001], FP-tree [Han et al., 2000], SPADE [Zaki, 2001], SLPMiner [Seno and Karypis, 2002] and GST [Huang and Lin, 2003]. These algorithms have been adapted for use with text data [Li et al., 2015, Wu, 2007, Zhong et al., 2012]; however, text patterns can still be redundant and noisy [Gao et al., 2015, Li et al., 2015]. Several pruning techniques, such as closed patterns [Yan et al., 2005], maximal patterns [Feldman et al., 1997] and master patterns [Yan et al., 2005], have been developed in the data-mining communities to remove noisy patterns and manage redundant patterns [Han et al., 2007, Mooney and Roddick, 2013, Xu et al., 2011]. These pruning techniques make closed sequential patterns as an alternative to phrases because they are (1) more frequent, (2) still possess some semantic information and (3) do not impose the strict rule of sequential occurrence of terms [Li et al., 2010, Wu, 2007]. However, extracting patterns from text data seems to be less efficient than term-based techniques [Algarni, 2014, Zhong et al., 2012], and discovering high-quality knowledge from patterns seems to impose further time complexity [Li et al., 2010, Wu, 2007, Zhong et al., 2012]. Moreover, informative patterns can suffer from the low-frequency problem if they are treated as a single atom [Wu, 2007, Zhong et al., 2012]. Also, the assumption that all terms in a closed sequential pattern are equally important to user information needs can be too simple and needs to be relaxed.

Selecting the most useful patterns for relevance discovery is challenging due to the large number of patterns generated from relevant documents using patterns' interestingness measures (i.e., supports and confidence) [Li et al., 2015, Yan et al., 2005]. Such selection may also lead to feature loss [Alharbi et al., 2017a,c]. Re-using minimum support and confidence in identifying relevance patterns from the broad set of discovered ones is not effective, because only statistical information about the used patterns is revealed [Li et al., 2010, Zhong et al., 2012]. For example, given a set of documents D that describes user information needs, patterns with low support values are generally short and highly frequent [Algarni, 2014, Li et al., 2015, 2008, Zhong et al., 2012]. Such patterns are more general and less specific to the topics discussed in D. Conversely, longer patterns have higher support and are more specific to the topics in D, but less frequent [Gao et al., 2015, Li et al., 2015]. However, identifying the best values for these measures is still experimental and hard to generalise, and adopting such measures for pattern-based TFS can make the model highly sensitive to them [Li et al., 2015, Zhong et al., 2012]. Moreover, textual patterns like phrases order terms based on their positional appearance in documents and do not arrange them according to relevance to the topics discussed in the documents, or even to what the user needs. Also, specifying the length of a pattern as a hyperparameter (i.e., how many terms a pattern must contain) is still beyond the user's capability, and generalising that longer or shorter patterns are always informative is rather too simple and difficult to justify.

To address some of these limitations, pattern deployment techniques, such as pattern deploying with relevance function (PDR) [Wu et al., 2006], pattern deploying method (PDM) [Zhong et al., 2012] and pattern deploying based on support (PDS) [Wu, 2007], have been developed and adopted to revise patterns extracted using the interestingness measures by first, finding some correlations between them, and then deploying (i.e., distributing) them to a hypothesis space (e.g., a positive or negative term space or both) [Li et al., 2004]. This technique has led to a significant performance in discovering relevant features based on text patterns [Algarni et al., 2010, Li et al., 2015, 2011]. However, we argue that this technique is still sensitive to the selected hypothesis space, which is generally just a BoW representation, and its statistical features' types, size, noise, redundancy, and generality and specificity of this space [Abul Bashar, 2017, Bashar et al., 2016]. Further, this space is hard for users to govern as it is usually the whole term's space of the D collection, in which it has been assumed that each term has the same importance (e.g., relevance) to each document. Such an assumption can be too simple, as terms tend to co-occur in every document $d \in D$ unevenly.

Also, pattern deployment cannot deal with ambiguous patterns that appear in negative feedback [Li et al., 2009b, Zhong et al., 2012]. Such patterns can influence the identification of relevant features [Li et al., 2010, 2012]. To deal with this issue and refine such problematic patterns, the pattern evolution technique [Wu, 2007, Zhong et al., 2012] has been introduced. Techniques that use patterns in negative feedback have shown considerable improvement in identifying relevant features. Popular examples are the pattern taxonomy model (PTM) [Wu et al., 2006, 2004], Relevance Feature Discovery (RFD$_1$) [Li et al., 2010] and RFD$_2$ [Li et al., 2015], which use different patterns and algorithms to reduce the side effects of high-level patterns and low-level terms that appear in both sets. However, these pattern-based models may not perform well if the positive and the negative feedback are mutually exclusive, as negative feedback can be any irrelevant documents. Pattern-mining algorithms also appear to suffer when the relevant documents are limited and collecting them is expensive and time-consuming [Algarni, 2011, Soleimani and Miller, 2016].

Concept-based Models

A concept is a set of semantically related words that together describe a human understanding of a particular object or idea [Bashar and Li, 2017, 2018, Egozi et al., 2008]. Concept-based TFS is supposed to effectively identify user information needs as concepts that reflect human understanding and knowledge of a particular topic [Bashar et al., 2017, Egozi et al., 2008]. However, concept-based models are sensitive to the type of text feature adopted to represent the agreeable concept as a set of related terms, whether these terms come from phrases [Liu et al., 2016], patterns [Bashar and Li, 2018, Bashar et al., 2017] or topics of topic modelling algorithms [Bashar and Li, 2017, Bashar et al., 2016]. Also, concept-based techniques can be manual, semi-manual or entirely dependent on external sources of knowledge [Tao, 2009] such as dictionaries (e.g., thesaurus) and domain ontologies, which can be incomplete or ambiguous. Moreover, the automatic specification of relevant and irrelevant concepts can be challenging, as only the user can decide based on the concept map in his or her mind after reading a retrieved document [Tao, 2009]. Simulating this process is difficult because manual concept specification implicitly lacks a clear understanding of the user's background knowledge [Tao, 2009]. Much attention has been paid to learning personalised ontologies from a set of documents that represents the user's profile to explicitly simulate user concept maps [Bashar et al., 2016, 2017, Shen et al., 2012a, Tao et al., 2011]. However, this process is challenging as it can be expensive, time-consuming and requires verification by domain experts [Bashar et al., 2016, Lee et al., 2015b, Li and Zhong, 2004, Zhu and Iglesias, 2017]. Constructing and updating ontologies for every knowledge domain is also unrealistic, especially considering the current exponential growth in data sources.

Ontological concepts can be general, incomplete and sensitive to the type of relations that govern the hierarchal structure between these concepts in the ontology (e.g., super-class, sub-class, is-a, part-of, etc.) [Bashar et al., 2016, Li and Zhong, 2004, Tao, 2009, Tao et al., 2011, Yuefeng and Ning, 2006]. Further, relying on ontologies to select informative features

[Bashar and Li, 2017, 2018, Bashar et al., 2016, 2017] can lead to feature loss, as no complete ontology can practically represent all existing human knowledge. Also, ontologies alone are not an accurate way to estimate feature weight or efficiently represent the relevance of features [Luo et al., 2011]. Concept-based TFS models endeavour to add more semantic knowledge to discovered features, such as terms [Egozi et al., 2008], phrases [Liu et al., 2016], patterns [Bashar and Li, 2018, Bashar et al., 2017] and topics of topic modelling algorithms [Bashar and Li, 2017, Bashar et al., 2016], using global knowledge base ontologies or those learned from local user repositories [Shen et al., 2012a,b, Tao et al., 2011]. However, while this approach can help humans interpret the meaning of these features, it still does not specify the features' importance to user information needs, especially when the user query is absent. A knowledge base ontology consists of a set of concepts with their semantic relations (e.g., is-a, part-of, related-to, etc.) that together represent a human background knowledge of a specific domain, or many sub-domains, of knowledge [Abul Bashar, 2017, Tao, 2009]. In the absence of a user-specific query, it is challenging to map user information needs that may be unevenly discussed in a set of relevant documents to a domain ontology (i.e., mapping many to many) [Abul Bashar, 2017]. It could introduce ambiguity and feature loss if the ontology is not comprehensive; further, such an approach is expensive and highly sensitive to the feature type, semantic relations and the ontology itself.

Overall, fusion-based TFS models that use terms, phrases, patterns, concepts or a combination of these, do not explicitly assume that a long document can exhibit multiple topics or themes [Gao et al., 2014b, 2015]; yet in reality, they can contain multiple semantically related topics or sub-topics [Anastasiu et al., 2013, Gao et al., 2015]. Topic-based models have been developed to address this assumption.

Topic-based Models

Probabilistic topic modelling algorithms, such as the probabilistic latent semantic analysis (PLSA) [Hofmann, 2001] and latent Dirichlet allocation (LDA) [Blei et al., 2003], have gained popularity and are widely accepted in IR, IF, NLP, TM and ML research communities [Blei, 2012, Gao et al., 2015, Wei and Croft, 2006, Xiong et al., 2015, Yi and Allan, 2009, Zhang and Chow, 2016]. These algorithms discover latent topics that can be used to represent user information needs [Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2017]. Existing TFS models that adopt terms, phrases, patterns, concepts or a combination of these, do not explicitly assume user information needs can discuss multiple topics [Alharbi et al., 2017b,c, Gao et al., 2014a]. For example, consider a researcher in forensic science who wants to enrich his or her knowledge about economic espionage by studying popular cases that have been identified worldwide. The two features (i.e., 'economic espionage') represent the researcher's information needs (e.g., a query to a search engine) and can be regarded as a phrase (bi-gram), two single terms or a pattern. However, searching algorithms that adopt these features do not consider that such an information need (i.e., the query 'economic espionage') can have multiple related topics or sub-topics, such as 'commercial espionage', 'industrial espionage', 'corporate espionage' and 'technical espionage' [Gao et al., 2017, 2015]. LDA is designed to consider this assumption automatically, but it favours the most frequent topics or sub-topics in the collection [Ding and Yan, 2015, Mimno et al., 2011]. Such algorithms can automatically, and in an unsupervised way, extract latent topics from pieces of text [Blei et al., 2003, Hofmann, 2001]. These topics are a reduced intermediate representation that can be used in a broad spectrum of applications and tasks [Gao et al., 2014b, 2015].

Unlike the PLSA, LDA is the most popular technique and its generated topic is a set of semantically related words [Blei et al., 2003, Gao et al., 2015]. LDA defines a topic as a probability distribution over all terms in the collection vocabulary [Blei, 2012, Blei et al., 2003], which (1) gives a user the ability to specify the length of a topic (i.e., how many terms it should have), (2) arranges topic terms based on their importance to the topic, and (3) relaxes the constraint of the strict sequential appearance of terms in a single topic. Further, LDA represents a document as a probabilistic mixture of multiple topics [Blei et al., 2003, Gao et al., 2015], which allows the location of similar topics or sub-topics across the collection and cluster documents or paragraphs that discuss similar subjects or themes [Anastasiu et al., 2013, Blei, 2012, Blei and Lafferty, 2009]. Despite these advantages, using LDA or PLSA for relevance discovery does not show encouraging performance because they cannot estimate a generalised (globally representative) weight for topical terms to reflect their relevance in a set of relevant documents that describe user information preferences [Gao et al., 2014b, 2015]. This is because these types of documents usually contain uncertainties that come from irrelevant parts in these documents, as users can label a document as relevant even though only a small part is relevant and the rest is irrelevant [Alharbi et al., 2018a, Bendersky and Kurland, 2010]. Therefore, estimating the relevance of features globally at the collection level can be effective if it is assumed that the relevant parts frequently appear across all relevant documents in the collection.

Mixed-Features Models

Mixed-based TFS models use different combinations of high-level features to overcome the limitations of single feature models or exploit the semantic information of two or more of these high-level features [Gao et al., 2015]. For example, n-grams and latent topics have been integrated into the topical n-grams model (TNG) [Wang et al., 2007] to discover topical phrases that are more discriminative and interpretable. Similarly, patterns and topics have been employed [Gao et al., 2017, 2014b, 2015] to take advantage of the explicit relationships between pattern terms and the multi-topics representation of documents in the topic modelling algorithms to produce a more discriminative and informative representation for user information needs. Further, ontological concepts have been used with patterns [Bashar and Li, 2018, Bashar et al., 2017] and latent topics [Bashar and Li, 2017, Bashar et al., 2016] to add explicit semantics to these features and facilitate the interpretation and understanding of their meanings. Yet, despite the advantages of mixed-based models, they can be time-expensive and susceptible to the previously mentioned inherited limitations of high-level features.

1.1.2.3 Hybrid Fusion Strategy

A hybrid fusion strategy can exploit the advantages of the early and late fusion approaches [Alqhtani et al., 2018, Atrey et al., 2010, Baltrušaitis et al., 2019]. Thus, a hybrid fusion based TFS model would integrate low-level terms with one or more high-level features. In relevance discovery, this hybrid strategy brings the statistical richness of individual terms in relevant documents to the semantic information of high-level features extracted from the same documents [Abul Bashar, 2017, Wu, 2007]. The advantage of this strategy can be clearly observed in the PDS [Wu et al., 2006, Zhong et al., 2012], RDF_1 [Li et al., 2010] and RDF_2 [Li et al., 2015] models, which map high-level patterns to low-level terms to solve the low-frequency problem of specific patterns. This strategy also made pattern mining a feasible technique for discovering relevant features that describe user information needs [Wu, 2007]. However, because text features can pass their limitations onto the models and frameworks that use them, adopting a mixture of low-level terms and high-level features can be ineffective as it might implicitly inherit the limitations of each feature type, especially if no solution for the limitations were provided beforehand [Alharbi et al., 2018a]. The critical problem in this issue is how to present, model and understand the complex relationships between these different types of features and relevant and/or irrelevant documents in the forms of weighting functions under one framework, and use them to discover or re-rank relevant features.

1.1.3 Text Feature Weighting Schemes

The weighting function is the most important component in a TFS model [Albathan et al., 2013, 2014, Li et al., 2015]. If this function fails in assigning the best and most representative weight to the feature, the whole fusion-based model will fail [Alharbi et al., 2018a]. Term- and phrase-based weighting functions are heuristic—even the probabilistic ones are frequency-based—and do not show effective performance in IF [Li et al., 2011, Zhong et al., 2012]. Pattern-based weighting functions are also heuristic and usually ignore the original semantic information (i.e., break the relationships between pattern terms) [Bashar et al., 2016, Zhong et al., 2012]. They are ineffective in accurately assigning a representative weight to terms that show the terms' relevance to the user information needs [Alharbi et al., 2018a]. Further, weighting functions do not address negative feedback. Existing supervised models, such as BM25 [Robertson and Zaragoza, 2009], SVM [Joachims, 2002] and Rocchio [Rocchio, 1971], are term-based and do not show sufficient performance as they ignore the multi-topics assumption [Gao et al., 2015, Li et al., 2015, 2010, 2012, Zhong et al., 2012].

A global weighting scheme can assign a representative weight to features because it is explicitly related to relevance judgements [Escalante et al., 2015, Greiff, 1998, Man et al., 2009, Shirakawa et al., 2015]. However, such schemes suffer from a lack of relevance details at the document level. Conversely, local weighting schemes estimate the relevance of features at the document level because a relevant document contains details about such relevance, but may be implicitly related to relevance judgements [Escalante et al., 2015, Greiff, 1998, Liu et al., 2009, Sabbah et al., 2017, Wu and Gu, 2017]. Therefore, these details are hard to use because they do not directly describe the available relevance at the collection level. Thus, the research issue is how to devise a middle solution that exploits the trade-offs between a global and local estimation of features relevance. Therefore, this research seeks to integrate early and late fusion strategies of different features by modelling complex relationships between the entities in a document collection (i.e., terms, topics, paragraphs, documents or clusters of documents) that share these features.

To do so, this research extends multiple random sets [Goutsias et al., 1997, Molchanov, 2005, Nguyen, 2008] to represent, describe and understand these complex relationships via multiple probabilistic functions. These functions are then effectively combined, globally and locally, to re-weigh topical terms based on their appearance across the selected entities. The proposed models and frameworks will be used to rank and select features for relevance discovery. They are application independent and can be applied to various tasks in text analysis. However, in this thesis, we tested the proposed models and frameworks for IF, which can be considered a special type of binary text classification [Gao et al., 2015, Li et al., 2008], as well as for ranking relevant terms (RRT) that were selected by domain experts.

1.1.4 Text Feature Selection Applications

Fusion-based TFS techniques have been experimentally evaluated for different TM, IR, IF and ML tasks [Forman, 2003, Metzler, 2007, Yang and Pedersen, 1997, Zhang et al., 2016]. In the absence of a query, which explicitly represents user information needs in IR, the proposed models and frameworks can be evaluated in the context of text classification and IF. However, text classification ignores relevant information ranking while IF is considered a binary text classification problem with more focus on the relevance ranking of information [Gao et al., 2015, Li et al., 2011, 2008]. As the name implies, an IF system removes information or documents from a stream of documents or information that do not meet user needs [Algarni et al., 2010, Belkin and Croft, 1992]. In this research, user information needs are represented by a set of related terms—used as a query in our case—that are discovered by the proposed models and frameworks from the set of documents in which the user is interested.

Based on the quality of the set of terms produced by the proposed research, the IF system is able to rank the most relevant documents that strongly meet the user's needs or interests. This research is evaluated empirically for IF and RRT. An extensive series of experiments have been conducted on the first 50 collections of documents of the standard Reuters Corpus Volume 1 (RCV1) dataset [Lewis et al., 2004], which are assessed by domain experts at the National Institute of Standards and Technology [1]. These collections imitate real user information needs, are high in terms of quality and reliability, and sufficient for a stable experiment [Buckley and Voorhees, 2000, Li et al., 2012]. The experimental results show that the proposed models and frameworks significantly outperform all baseline models, regardless of the type of text feature they adopt, the fusion strategies they apply or the learning algorithms they use.

[1] https://www.nist.gov/

1.2 Problem Statement and Objectives

The previous section introduced extensive background knowledge on various TFS techniques from different perspectives, including data fusion, and generally described their apparent limitations in identifying relevant features that can be used to represent user information needs. The discussion motivates the research work in this thesis by pinpointing the need for more effective fusion-based techniques for relevant feature discovery. The following section lists and discusses the main research questions addressed in this thesis and their objectives.

1.2.1 Research Problems

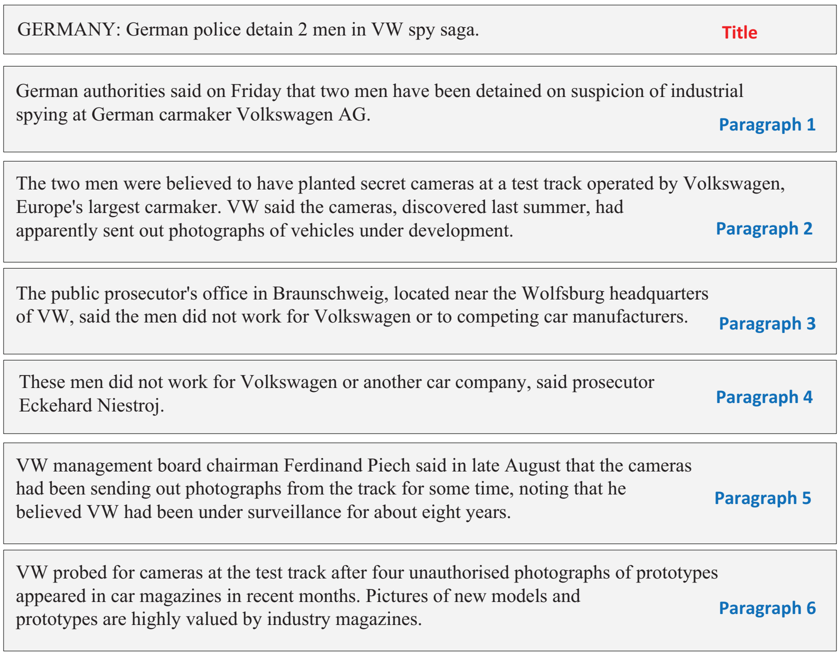
Probabilistic topic modelling algorithms such as PLSA [Hofmann, 2001] and LDA [Blei et al., 2003] are widely researched and broadly applied [Blei et al., 2010a, Blei, 2012, Griffiths and Steyvers, 2004, Wei and Croft, 2006]. Most research in topic modelling addresses the issues of efficiency and scalability of the algorithms, and the interpretation, semantics and cohesion of generated topics [Bashar and Li, 2017, Chuang et al., 2013, Gao et al., 2017, He et al., 2017, Ramage et al., 2011]. However, searching for useful and relevant topical features is an ongoing research problem, as demonstrated by some very recent studies [Ma et al., 2019, Wu et al., 2019, Xu et al., 2019].

In TFS, different types of text features are used to represent user information needs discussed in a set of documents. While terms, phrases, patterns, concepts or a combination of these do not assume user information needs can exhibit multiple topics or sub-topics [Gao et al., 2013, 2014a, 2015], latent statistical topics discovered by topic modelling techniques are explicitly built on the assumption that a document can discuss multiple topics [Blei et al., 2003, Deerwester et al., 1990, Hofmann, 2001], which makes them more representative of what the user needs [Gao et al., 2014b, 2015, Wu et al., 2019]. As a set of semantically related terms sorted in descending order based on their importance to the topic, a topic can alleviate the problem of polysemy and synonymy to some extent by softly clustering similar words together in the form of a topic [Blei et al., 2003, Hofmann, 2001]. Also, topic modelling can reduce the dimensions of a text corpus to a set of a limited number of topics [Gao et al., 2014b, 2015]. However, as previously noted, topic modelling algorithms do not show encouraging results for relevant feature discovery. Nevertheless, given all the advantages of topic modelling, the primary problem concerns effectively utilising the discovered topics for selecting relevant features.

Fusion-based IR models that adopt data fusion or collection fusion have shown remarkable results compared to traditional techniques in the field [Lillis et al., 2006, 2008, 2010, Nuray and Can, 2006, Towell et al., 1995]. Existing research demonstrates that fusing different representations of documents, queries, search results, rankings and scores can lead to substantial improvements on single IR models [Anava et al., 2016, Croft, 2000, Kozorovitsky and Kurland, 2011a,b, Pickens and Golovchinsky, 2008, Zhang and Balog, 2017]. However, applying similar fusion techniques to TFS for relevance discovery under uncertainties is limited, as no single text feature can encompass information relevant to the user needs, which makes the feature fusion strategies more prominent. Moreover, there is no similar technique in current relevance discovery literature that models the fusion of different hierarchal features and integrates multiple relevance fusion models into supervised and unsupervised frameworks for relevant feature selection. This research gap is the basis of research questions that will be answered in this thesis.

As previously discussed, statistical topic modelling algorithms reduce the dimensions of a text corpus to a specified set of topics. Each topic groups semantically related terms together as they appear in the corpus, which alleviates the problems of synonymy and polysemy to a certain extent [Hofmann, 2001, Steyvers and Griffiths, 2007]. The algorithms also assume that each document can discuss multiple topics to imitate this reality, especially long documents (see Figure 1.1), in which different themes tend to be discussed across relevant document paragraphs. This assumption makes these algorithms more capable of identifying the hidden needs of users [Gao et al., 2014b, 2015]. Figure 1.2 illustrates a real example of hidden needs from Collection 101 of the RCV1 dataset, which is about ‘economic espionage’. The narrative element in the figure clearly shows the relevant themes or sub-topics of this type of espionage. However, in the absence of user queries, such algorithms do not have an implicit mechanism to discover the most relevant features because they cannot accurately generalise the topical term weight to a more representative global level, especially when the term has an identical meaning (i.e., semantically the same) in a group of similar documents. Therefore, the first research problem to be addressed is:

- **RQ1**: How do we effectively fuse different features from a collection of documents that describes user information needs to accurately generalise topical term weight globally at the collection level?



GERMANY: German police detain 2 men in VW spy saga.

Title

German authorities said on Friday that two men have been detained on suspicion of industrial spying at German carmaker Volkswagen AG.

Paragraph 1

The two men were believed to have planted secret cameras at a test track operated by Volkswagen, Europe's largest carmaker. VW said the cameras, discovered last summer, had apparently sent out photographs of vehicles under development.

Paragraph 2

The public prosecutor's office in Braunschweig, located near the Wolfsburg headquarters of VW, said the men did not work for Volkswagen or to competing car manufacturers.

Paragraph 3

These men did not work for Volkswagen or another car company, said prosecutor Eckehard Niestroj.

Paragraph 4

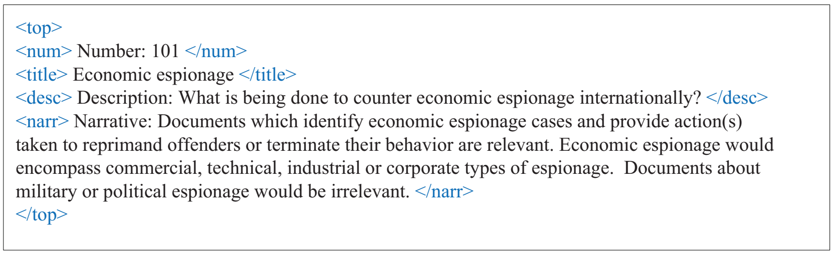
VW management board chairman Ferdinand Piech said in late August that the cameras had been sending out photographs from the track for some time, noting that he believed VW had been under surveillance for about eight years.

Paragraph 5

VW probed for cameras at the test track after four unauthorised photographs of prototypes appeared in car magazines in recent months. Pictures of new models and prototypes are highly valued by industry magazines.

Paragraph 6

Figure 1.1: A sample of a relevant long document from collection 101 of the RCV1 dataset.



<top>
<num> Number: 101 </num>
<title> Economic espionage </title>
<desc> Description: What is being done to counter economic espionage internationally? </desc>
<narr> Narrative: Documents which identify economic espionage cases and provide action(s) taken to reprimand offenders or terminate their behavior are relevant. Economic espionage would encompass commercial, technical, industrial or corporate types of espionage. Documents about military or political espionage would be irrelevant. </narr>
</top>

Figure 1.2: A TREC topic for collection 101 of the RCV1 dataset in which the *title* element '*Economic espionage*' represents explicit user information needs.

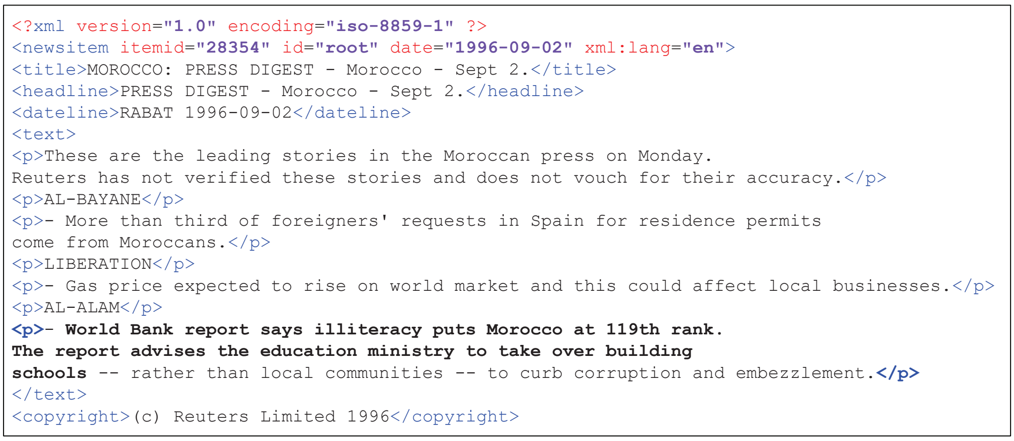
Assuming a term has equal importance in a group of similar documents that describe user information needs can be a simple assumption. Global (i.e., corpus level or collection level) feature selection methods have adopted such an assumption by estimating the importance (i.e., a relevance weight) of the term based on its global information in the whole corpus, in a heuristic way [Chen et al., 2016, Cummins and O'Riordan, 2006, Shirakawa et al., 2015]. Local methods have tried to relax the constraint of a globally generalised term weight by considering its importance locally, in a document-by-document manner [Chen et al., 2016, Sabbah et al., 2017]. However, both approaches—especially those based on term, phrase, concept and pattern methods—do not assume a document can exhibit multiple topics or themes even though in reality, a long document can span different sub-topics in its segments (i.e., its paragraphs or sentences), as illustrated in Figure 1.1. Topic-based techniques such as PLSA [Hofmann, 2001] and LDA [Blei et al., 2003] can be considered local TFS methods that have adopted multi-topics representation of documents.

Yet, despite relaxing the term global assumption, these topic-based methods do not show encouraging performances in identifying relevant features [Alharbi et al., 2018a, 2017a,b,c, 2018b, Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2015]. From a data fusion perspective, this is because they estimate the local importance of terms based on the fusion of two topical features: the document topic and term topic probability distributions [Blei et al., 2003]. The former is flat and does not automatically consider the hierarchal sub-features of the document (e.g., its paragraph topic features), as both PLSA and LDA use BoW representation and do not retain the notion of a document [Blei, 2012, Wallach, 2006, Wang et al., 2007]. The latter is globally estimated from the entire corpus, which makes it sensitive to frequency as well as uncertainties in relevance feedback. Therefore, to develop a more effective fusion-based TFS for relevance discovery, the following research problem will also be addressed in this thesis:

- **RQ2**: How do we effectively fuse local and global features to more accurately estimate the generalised topical term weight?

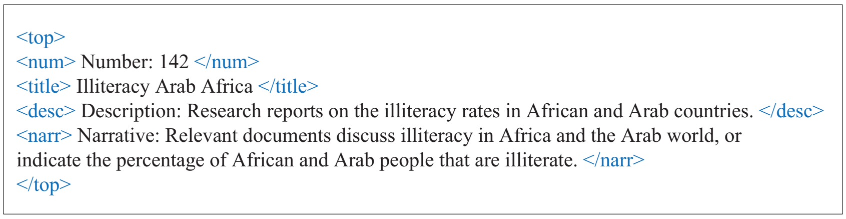
For most relevance discovery models, regardless of the fusion strategy, feature type or learning and mining algorithm used, the document level is an evidence space for identifying relevant features to represent user information needs [Gao et al., 2017, 2014b, 2015]. As previously mentioned, a document can be labelled as relevant even if only a small part of it contains relevant information [Bendersky and Kurland, 2010, Fan et al., 2018, Kaszkiel and Zobel, 1997, Liu and Croft, 2002]. This is demonstrated in the real example of a labelled relevant document from collection 142 of the RCV1 dataset shown in Figure 1.3, as only a small segment of one particular paragraph is considered relevant based on the TREC topic description of the collection (see Figure 1.4). Thus, selecting features from all parts of such a document leads to uncertainties and scatters the focus on relevant information because features from non-relevant parts do not represent user information needs [Alharbi et al., 2018a, Lv and Zhai, 2010]. Consequently, the relevance of the corresponding part should be considered when selecting features from it. Research in IR shows that considering the evidence at the passage level (e.g., a paragraph level) can improve retrieval accuracy, especially when documents are long or span different subject areas [Bendersky and Kurland, 2010, Callan, 1994, Dang et al., 2015, Liu and Croft, 2002]. However, in TFS for relevance discovery and in the absence of an explicit query representing user needs, which can also guide the search for a relevant paragraph, it becomes very challenging to explicitly estimate a paragraph's relevance in a set of documents that describe user information needs. Therefore, an implicit mechanism is needed to utilise the paragraph level evidence. To reduce uncertainty in the relevant feature discovery, we address the following research problem in this thesis:

- **RQ3**: How do we effectively fuse multiple features in a collection of relevant documents to implicitly estimate the paragraph relevance and use it to manage uncertainties in relevant features discovered by existing TFS models?



```
<?xml version="1.0" encoding="iso-8859-1" ?>
<newsitem itemid="28354" id="root" date="1996-09-02" xml:lang="en">
<title>MOROCCO: PRESS DIGEST - Morocco - Sept 2.</title>
<headline>PRESS DIGEST - Morocco - Sept 2.</headline>
<dateline>RABAT 1996-09-02</dateline>
<text>
<p>These are the leading stories in the Moroccan press on Monday.
Reuters has not verified these stories and does not vouch for their accuracy.</p>
<p>AL-BAYANE</p>
<p>- More than third of foreigners' requests in Spain for residence permits
come from Moroccans.</p>
<p>LIBERATION</p>
<p>- Gas price expected to rise on world market and this could affect local businesses.</p>
<p>AL-ALAM</p>
<p>- World Bank report says illiteracy puts Morocco at 119th rank.
The report advises the education ministry to take over building
schools -- rather than local communities -- to curb corruption and embezzlement.</p>
</text>
<copyright>(c) Reuters Limited 1996</copyright>
```

Figure 1.3: A sample of a relevant long document from collection 142 of the RCV1 dataset that discusses '*Illiteracy Arab Africa*' as shown in Figure 1.4 with only a part of a paragraph is considered relevant (the last paragraph).



```
<top>
<num> Number: 142 </num>
<title> Illiteracy Arab Africa </title>
<desc> Description: Research reports on the illiteracy rates in African and Arab countries. </desc>
<narr> Narrative: Relevant documents discuss illiteracy in Africa and the Arab world, or
indicate the percentage of African and Arab people that are illiterate. </narr>
</top>
```

Figure 1.4: A TREC topic description for collection 142 of the RCV1 dataset in which the *title* element '*Illiteracy Arab Africa*' represents explicit user information needs.

In addition to the research problems mentioned, unsupervised topic modelling algorithms seem to favour subjects that frequently appear in a text corpus [Ding and Yan, 2015, Mimno et al., 2011, Xu et al., 2019]. These highly frequent subjects can overshadow less frequent but equally important subjects, especially in relevance discovery. Given a set of documents relevant to user needs, LDA assigns higher probabilities (i.e., weights) to features that frequently appear within a limited subset of documents while those occurring in fewer documents receive less attention, even though they may be equally relevant to user needs. Document clustering is also an unsupervised learning algorithm that appears to effectively limit bias towards highly frequent subjects by grouping documents that share common subjects in a hard cluster [Aggarwal and Zhai, 2012, Anastasiu et al., 2013, Jain, 2010]. However, a traditional clustering algorithm does not consider the detailed topics or themes exhibited across documents, and assuming a cluster of similar documents only discusses one topic is too simple [Alharbi et al., 2017b, Krikon and Kurland, 2011, Liu and Croft, 2004]. Further, selecting informative features from a set of equally important clusters and assigning globally representative weights to these features is difficult when no search guide (e.g., a user query) is given. Combining these issues with the previous research problems, especially uncertainties in relevance feedback, this thesis raises the following research question:

- **RQ4**: How do we effectively develop an unsupervised relevance discovery framework by integrating topic modelling, document clustering and global statistics to effectively select, weigh and ultimately fuse different intra- and inter-cluster relevant features?

Discovering relevant features from a collection of relevant and irrelevant documents to specifically describe what the user needs remains a major research problem in IR, IF, DM and ML communities [Gao et al., 2015, Li et al., 2015, Man et al., 2009, Yuefeng and Ning, 2006], as it is both theoretically and empirically challenging [Li et al., 2015, 2010]. From the data fusion perspective, unsupervised TFS techniques that adopt early, late or both fusion strategies are not discriminating enough to accurately weigh features that frequently appear in both relevant and irrelevant documents [Hou et al., 2010, Man et al., 2009]. Supervised techniques can be discriminative in selecting specific features of the class label that separate relevant and irrelevant documents, but because they do not consider the detailed latent structures of these labelled documents, they cannot assign globally representative weights to show features' relevance to user needs [Alharbi et al., 2018b, Gao et al., 2015, Li et al., 2015]. Considering the research problems of unsupervised topic modelling and the issues of feature weighting in supervised relevance discovery models, as well as the uncertainties in relevance feedback, the following research question is posed:

- **RQ5**: How do we effectively develop a supervised relevance discovery framework by integrating discriminative learning algorithms, topic modelling and global statistics and effectively fuse both discriminative and descriptive features to identify relevant features that more specifically represent user information needs?

To solve these five research questions, the thesis research objectives are discussed in the following section.

1.2.2 Research Objectives

The main objective of the research in this thesis is to develop better TFS models and frameworks for relevance discovery. The selected features and their weights must be informative and representative to user information needs. Therefore, the research outcomes are not targeting any specific areas of applications and should improve applicable information-oriented systems, such as IF, IR, text classification and others. The research contributions made in this thesis are original and highly significant, especially in the field of relevant feature discovery in TM.

The research work in this thesis solves the identified problems of topic modelling and brings data fusion techniques into the area of relevance discovery for more effective TFS models and frameworks. Unlike common practice in data fusion, whereby multimodal data sources are integrated to produce more accurate, consistent and useful information, this research only depends on different text features from a single collection of documents. The collection is domain-specific and has a relatively small set of relevant and irrelevant documents that can be used to discover user information needs. Features are extracted from the collection using some supervised and unsupervised learning algorithms, namely topic modelling, clustering and SVM, including some global and local statistical features. No external sources of knowledge are used in this research.

- **RO1**: To accurately estimate a more globally representative weight for a topical term, a hybrid fusion based model (called SIF) is proposed in which multiple random sets are extended to (1) manage a hybrid fusion of features of distinct entities in the document collection and (2) model the complex relationships between these features and the entities that influence the term weighting process.

Adopting a hybrid fusion strategy guarantees that the fusion is based on some semantic information (i.e., extracted by topic modelling) rather than based solely on heuristics. The fusion is between different features of the collection's paragraphs, the latent topics extracted from these paragraphs and all terms in the collection (i.e., the vocabulary list). As the associations between these entities are complex—in the form of many-to-one and one-to-many relationships—and topical terms frequently appear across numerous documents, paragraphs and even topics in the collection, it is challenging to know which relationship or entity is most important. Therefore, in the proposed SIF model, multiple random sets are extended and used to represent and thus, understand these relationships using probability functions. The fusion made by these functions can effectively estimate a more representative and generalised topical term weight, as will be fully described and evaluated in Chapters 3 and 6, respectively.

- **RO2**: To relax the assumption of the globally generalised topical term weight, the SIF model will be revisited and an integration between early and late fusion strategies of local and global features will be modelled using multiple extended random sets (ERS).

The proposed SIF model adopts a hybrid fusion strategy to select informative features for relevance discovery. It is a collection-based model that assumes identical topical terms have equal importance in every document in the corpus. Such an assumption can be too simple and needs to be accurately relaxed. In Chapter 4, the SIF2 model is introduced to solve the issue of the SIF model. The proposed SIF2 model adopts a hybrid fusion strategy of local and global features that are modelled by ERS. Unlike SIF, four entities and their complex relationships are represented and probabilistically estimated on a document-by-document basis to accurately measure the topical term importance in each document independently. As each document is equally relevant to the user's needs, it is difficult to identify the most representative weight for a topical term that is independently estimated in each document of the collection. To solve this problem, SIF2 assumes all individual weights of a topical term are important and combines them with a more descriptive global statistic. In addition to Chapter 4 presenting the details of the proposed SIF2 model, Chapter 6 shows an extensive experimental evaluation for this model.

- **RO3**: To reduce uncertainties in relevant features discovered by existing TFS models, an uncertainty reduction (UR) method is proposed to implicitly estimate the paragraph level evidence of relevance and use it to re-rank the discovered features after scaling their original weights.

Generally, the 'relevance' of a text feature in this thesis refers to the relevancy between a document or set of documents in which this feature is discovered and user information needs, which are supposed to be implicitly described across the contents of the document(s). Most existing relevance models treat all document segments (e.g., paragraphs and sentences) equally when discovering and weighing relevant features. However, this approach can introduce uncertainties to these features. The simple intuition behind this reason is that document segments are not equal in terms of their relevance to what the user needs and some can be completely irrelevant. Thus, considering non-relevant or weakly relevant parts when discovering and weighing relevant features can do more harm than good.

Numerous studies in IR confirm that adopting passage level evidence (e.g., paragraph level) for relevance shows remarkable improvements for query-based models in different retrieval tasks. However, in relevant feature discovery, such an explicit query can be either unavailable or not considered, which makes this problem particularly challenging. Therefore, in Chapter 4, a UR method is also proposed to implicitly estimate the relevance of a document's paragraph by modelling the late fusion of different features of the paragraph with latent topics and the paragraph with terms. Multiple ERS with inverses are developed to map, measure and understand the relationships between the four entities. Based on the proposed ERS theory, a feature weighting formula is developed to scale and re-rank the relevant features discovered by existing TFS models. The extensive evaluation of the proposed method is presented in Chapter 6.

- **RO4**: To force topic modelling algorithms to pay equal attention to both frequent and less frequent relevant topics of interest in an unsupervised way, and to select and weigh relevant features of these topics, an unsupervised relevance discovery framework of hybrid fusions is proposed.

To limit the topic modelling bias towards highly frequent subjects, a document-clustering technique will be employed to group documents that share similar subjects in one cluster. However, in IR it was assumed that clustered documents only discuss one subject [Alharbi et al., 2017b, Krikon and Kurland, 2011, Liu and Croft, 2004]; yet, such an assumption can be unrealistic, as a single long document in a cluster can exhibit multiple topics or sub-topics. Therefore, the clustering technique cannot reveal the detailed topical structures of documents and provides no clear way to either select or weigh the inter- or intra-cluster features. Conversely, topic modelling is capable of such a task and has been developed on the assumption that a document can have multiple topics. Nevertheless and as previously noted, topic modelling still (1) suffers from its inability to generalise topic term weights to a global level, (2) does not consider paragraph level evidence (as it is a document-wide model) and (3) pays no explicit attention to the document's hierarchal features when estimating weight.

Therefore, in Chapter 5, an unsupervised, two-stage, hybrid fusion based framework, called unsupervised selection of informative features (USIF) is proposed to discover relevant features that represent user information needs. The framework effectively integrates global statistics, topic modelling and document clustering to select and re-weigh clustered features. Multiple ERS are also developed to model the integrated hybrid fusions of multiple cluster-based and collection-based features and to describe and thus, understand the complex relationships between them. The idea of concept agglomeration is introduced in this framework to effectively identify the relevant inter-cluster features. The SIF model and an adapted version of the UR method are employed to estimate the topical and thematic significances of terms in the collection, respectively. These two significances will be used to discover the intra-cluster relevant features. The extensive experimental evaluation for the proposed framework is reported in Chapter 6.

- **RO5**: To discover and accurately weigh specific relevant features using both relevant and irrelevant documents, a supervised relevance discovery framework of hybrid fusions is proposed.

To reduce the impact of relevant features that frequently appear in both positive and negative training documents, a discriminative supervised learning algorithm will be used (e.g., SVM) to delineate between positive and negative documents. The boundary (e.g., the hyperplane as in SVM) is then used to select some discriminative specific features. However, such algorithms can implicitly inherit the limitations of the text feature they use and the uncertainties available in the training documents. In general, these algorithms do not consider the hidden topical structure of a training relevant document and have no explicit mechanism in adopting paragraph level evidence of relevance as a means to deal with the uncertainties. These reasons make them ineffective in assigning a more representative weight to the specific feature they discover. As already discussed, probabilistic topic modelling are unsupervised algorithms that effectively reveal the internal topical structure of the document, but they remain problematic.

SIF, SIF2 and the UR method have effectively solved these problems in a domain-specific context. However, as they are also unsupervised, they cannot deal with features that frequently exist in both negative and positive documents, as negative training documents, specifically, can be much larger and topically diverse. In Chapter 5, a two-stage supervised and hybrid fusion based framework, called supervised selection of informative features (SSIF) is proposed to specifically deal with negative documents and enhance the discovery of relevant features that represent user information needs. The proposed SSIF framework effectively incorporates global statistics, topic modelling and SVM to select and re-weigh discriminative specific features. A multiple hybrid fusions strategy is adopted in SSIF, which is modelled by multiple ERS, as in the USIF framework. Chapter 6 presents an extensive experimental evaluation of SSIF.

1.3 Contributions

The research work in this thesis contributes theoretically and practically to the field of TFS for relevance discovery. The contributions are original and significant and are implemented in the forms of different models (SIF, SIF2 and the UR method) and frameworks (USIF and SSIF) of TFS. The research integrates supervised and unsupervised learning algorithms and adopts data fusion strategies to select, weigh (or re-weigh) and rank (or re-rank) relevant features. A novel ERS theory is developed to model the integrations and manage the fusion of different local and global features. Several accurate feature-weighing schemes are also proposed based on the ERS modelling. The extensive experimental evaluation shows that the proposed models and frameworks are effective and significantly outperform all state-of-the-art baseline models regardless of the text features or the fusion strategies they utilise. More details on the contributions of this thesis are provided in Section 7.2.

The main contributions of this thesis to TFS for relevance discovery research are summarised as follows:

- An innovative hybrid fusion based model that extends multiple random sets to generalise the weight of topical terms in relevant documents based on a new and accurate term weighting scheme.

- A new and effective ERS-based model that integrates early and late fusion strategies to relax the assumption of a generalised weight of a topical term.

- An innovative and effective fusion-based method that adopts paragraph level evidence, at both the document and collection level, to reduce the uncertainty in relevant features discovered by existing TFS models.

- A new and effective unsupervised framework that integrates topic modelling, global statistics and document clustering to select and accurately re-weigh relevant features.

- An effective and new supervised framework that combines topic modelling, global statistics and SVM to select and accurately re-weigh relevance-specific features using both relevant and non-relevant documents.

1.4 Research Methodology

The research methodology can be defined as the theoretical framework through which researchers analyse the method or set of methods that are applied to solve the identified research problem [Gable, 1994, Leedy and Ormrod, 2005]. Scientific [Galliers, 1992], case study [Gable, 1994], action research [Somekh, 2005] and prototyping [Creswell, 2013] are examples of research methodologies that are applicable to the field of knowledge discovery in texts [Wu, 2007]. As this research aims to contribute to the knowledge discovery in text field, this makes it an empirical research type that is applicable to the scientific research methodology. However, as our proposed research has different stages and the scientific methodology consists of six repetitive activities, an organising methodology such as action research is needed. Therefore, after analysing all aspects of this research and the scientific and action research methodologies, we found the integration of scientific and action research methodologies best fit this research.

To achieve the aims of this thesis and solve the identified research problems, extensive surveys are conducted against the relevant literature of TFS, topic modelling, data fusion, measuring uncertainty, TM, document clustering, text classification and IF and retrieval. Then, a hypothesis is developed for each problem and an initial solution is proposed by developing a theoretically sophisticated model. Next, an experiment is designed to test the hypothesis and the initial results are evaluated. If the results are not significantly better than state-of-the-art baseline models, then iterative steps are taken until a better solution is achieved. These steps are revising the literature, updating the hypothesis, improving the proposed model, and testing and evaluating the model. Figure 1.5 illustrates the research approach used in this thesis.

1.5 Publications

Some parts of the proposed models and frameworks in this thesis and their results have been published in (or submitted to) international conferences and journals as follows:

Peer-Reviewed Journal Articles

- Abdullah Alharbi, Md Abul Bashar, Yuefeng Li, 'Fusing clustering and topic modelling for unsupervised relevant feature discovery', *IEEE Trans. Pattern Anal. Mach. Intell.* (**To be submitted**).

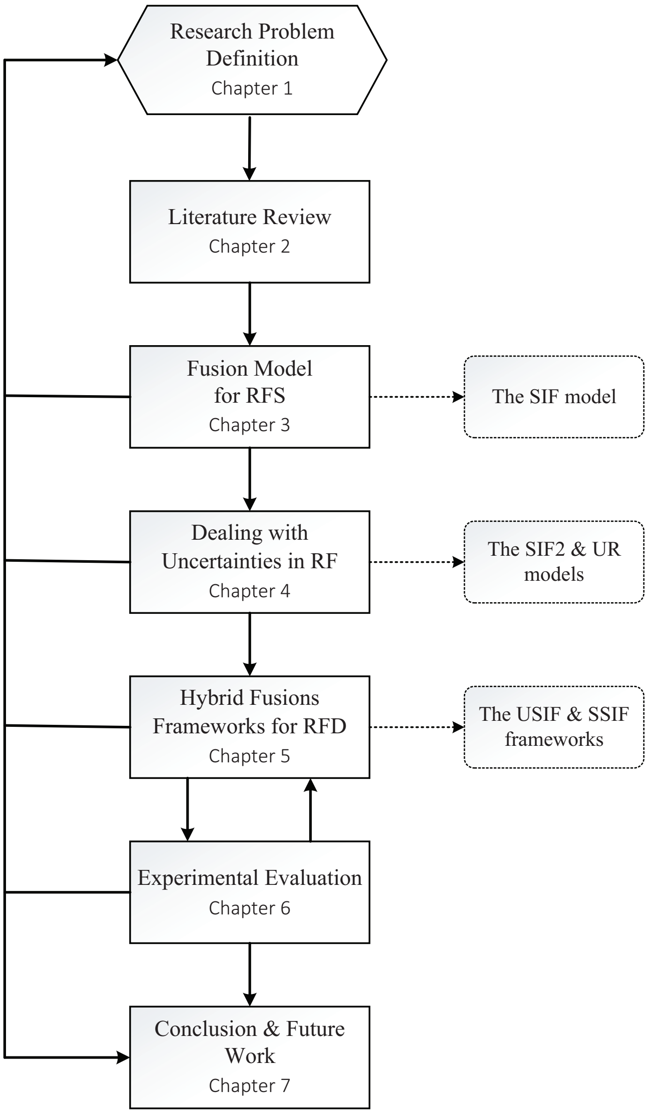


Figure 1.5: Research methodology and thesis structure.

- Abdullah Alharbi, Md Abul Bashar, Yuefeng Li, 'Combining supervised and unsupervised learning for an effective representation of specific corpus', *IEEE Trans. Knowl. Data Eng.* (**To be submitted**).

Peer-Reviewed Full Conference Papers

- Abdullah Alharbi, Yuefeng Li and Yue Xu, 'Enhancing topical word semantic for relevance feature selection', in *Proc. IJCAI Workshop on Semantic Machine Learning*, Melbourne, (vol. 1986), 2017, pp. 27–33.

- Abdullah Alharbi, Yuefeng Li and Yue Xu, 'Integrating LDA with clustering technique for relevance feature selection', in Peng W., Alahakoon D., Li X. (eds) *AI 2017: Advances in artificial intelligence*. Lecture notes in computer science. Cham: Springer, 2017, vol.

10400, pp. 274–286.

- Abdullah Alharbi, Yuefeng Li and Yue Xu, ‘Topical term weighting based on extended random sets for relevance feature selection’, in *Proc. Intern. Conf. on Web Intelligence*, Leipzig, Germany, 2017, pp. 654–661.

- Abdullah Alharbi, Yuefeng Li and Yue Xu, ‘An extended random-sets model for fusion-based text feature selection’, in Phung D., Tseng V., Webb G., Ho B., Ganji M., Rashidi L. (eds) *Advances in knowledge discovery and data mining. Lecture notes in computer science*. Cham: Springer, 2018, vol 10939, pp. 126–138.

- Abdullah Alharbi, Md Abul Bashar and Yuefeng Li, ‘Random-sets for dealing with uncertainties in relevance feature’, in Li X., Mitrovic T., and Xue B. (eds) *AI 2018: Advances in artificial intelligence. Lecture notes in computer science*. Cham: Springer, 2018, vol 11320, pp. 656–668.

1.6 Thesis Structure

This thesis is organised into seven chapters, as illustrated in Figure 1.5 and summarised as follows:

- **Chapter 2**: This chapter is a literature review of disciplines related to knowledge discovery in databases, including text mining. It comprehensively reviews and critically discusses recent research in TFS; specifically, research from different perspectives, including TFS applications. Limitations are pointed out, and possible solutions are suggested.

- **Chapter 3**: An innovative hybrid fusion based model for relevant feature selection (called SIF) is presented in this chapter. The model is unsupervised and proposed to address the limitations of the topical features discussed earlier in this chapter and Chapter 2. The model extends multiple random sets to describe the complex relationships between topical terms and other entities in a document collection. Based on these relationships, a weighting scheme is developed to estimate a generalised term score that effectively reflects the relevance of a term to user information needs and maintains the same semantic meaning of terms across all relevant documents.

- **Chapter 4**: This chapter describes SIF2, another novel, unsupervised fusion-based model for selecting informative features in a collection of documents that discusses a specific topic of interest that describes user information needs. SIF2 relaxes the term weight generalisation assumption of the SIF model, which has been explained in Chapter 3. It also adopts the strategy of distributing global term topics assignments generated by LDA to a local hypothesis space to solve unbalanced frequency-related problems. This chapter also presents the UR method, which is proposed to reduce uncertainties in relevant features discovered by existing TFS models of relevance discovery. The UR method is also unsupervised and estimates the relevance evidence in passages of relevant documents to scale the weight of these features and re-rank them accordingly.

- **Chapter 5**: Two novel frameworks of fusion-based TFS for relevance discovery are presented in this chapter. The first proposed framework, USIF, is unsupervised and integrates document clustering, topic modelling, concept agglomeration and global statistics in a two-stage approach to select and then re-weigh relevant features in the text collection that describe a specific topic of interest that discusses user information preferences. The second framework proposed is SSIF. It is a supervised, two-stage framework that combines SVM linearly with topic modelling and global statistics. It can effectively deal with the impact of topical terms that commonly appear in relevant and irrelevant documents.

- **Chapter 6**: The evaluation methodology for the proposed models and frameworks are detailed in this chapter, which includes evaluation hypotheses, the benchmark dataset, experimental design, evaluation measures and baseline models and their settings. A detailed analysis and discussion of the experimental results for each proposed technique in IF and RRT are also presented.

- **Chapter 7**: This concluding chapter summarises the key outcomes and discusses the significant contributions of the thesis. Identified limitations are also reported in this chapter, suggesting the direction of further research work in the future.

Chapter 2

Literature Review

This chapter undertakes a literature review of TFS techniques. The review is organised around the major areas of TFS to ensure that an exhaustive approach is taken. The first part covers knowledge discovery in databases in which feature selection is an essential pre-processing step. This section also describes strategies of text mining to extract representative text features. The second part of the review discusses the idea of TFS and proposes a taxonomic model to organise the study of existing TFS models. While this study focuses on TFS from the data fusion perspective, three other viewpoints are introduced in this chapter—namely, the search strategy of the TFS models, the availability of semantic information in the utilised features, and the models need for labelled training documents. Each of these perspectives is presented in a separate section, and popular TFS models are described and critically discussed under the relevant category. In an independent section, the applications of TFS are explained and linked to the proposed TFS taxonomy as each application requires a suitable TFS method. The last section presents a summary of this chapter.

As extensively discussed in Chapter 1, this study brought the feature fusion approaches to TFS for discovering relevant features that reflect user information needs. However, in the literature, feature fusion strategies are mainly used to exploit multimodal data more than the monomodal text. Therefore, we explicitly defined and critically discussed the concept of text feature fusion in Chapter 3 instead of this chapter. Additionally, based on the research studies reviewed in this chapter, it is clear that there is a research gap on how feature fusion can be used to combat uncertainties that affect most TFS models and frameworks. This study aims to fill this research gap.

2.1 Knowledge Discovery in Databases

Knowledge discovery is commonly defined as the process of extracting useful information from large databases of particular interest to specific users [Fayyad et al., 1996, Frawley et al., 1992]. The extraction process should be non-trivial, and the discovered information must: (1) implicitly exist in the databases, (2) be previously unknown to the users and (3) meet the users' information needs [Frawley et al., 1992]. From a formal point of view, knowledge discovery can also be defined as follows: if F is a set of facts that represent the given data, L is a language, and C is a set of certainty measures, then a statement $S \in L$ is called a pattern, which is discovered from the subset F_s taken from F based on a certainty c [Wu, 2007]. Thus, the pattern S must describe the relationships between the subset F_s of F and must be simpler than the listing of all facts in F_s. In this case, the pattern can be called knowledge if it meets the interestingness measures and certainty criteria imposed by the user. However, patterns with insufficient certainties cannot be treated as knowledge because the certainty with an acceptable degree is essential in the knowledge discovery process [Wu, 2007].

Overall, a knowledge discovery system must output high-quality patterns that demonstrate the following characteristics [Fayyad et al., 1996, Wu, 2007]

- **Interestingness**: On the basis that patterns must be novel, useful, and discovered in a non-trivial way, it implies that the discovered knowledge must be interesting too.

- **Accuracy**: Based on the used certainty measure, the discovered patterns should accurately reflect the contents of the original dataset in which inaccurate reflection should be noted by the certainty measure.

- **Efficiency**: Given a large database, this characteristic implies that the knowledge discovery algorithms must be efficient in terms of run-time where efficient means the algorithm run-time is acceptable and can be theoretically predicted.

- **Understandability**: This characteristic emphasis that a discovered knowledge must be interpretable using a high-level language. Thus, users should be able to understand this interpretation.

2.1.1 Knowledge Discovery Processes

Knowledge discovery is not a monolithic task. It involves several iterative and interactive processes, as shown in the model in Figure 2.1. These processes are data selection, data pre-processing, data transformation, data mining and pattern evaluation or interpretation, and each process depends on the output of the previous one [Fayyad et al., 1996, Frawley et al., 1992, Wu, 2007], as illustrated in the figure. Each of these processes is briefly described below.

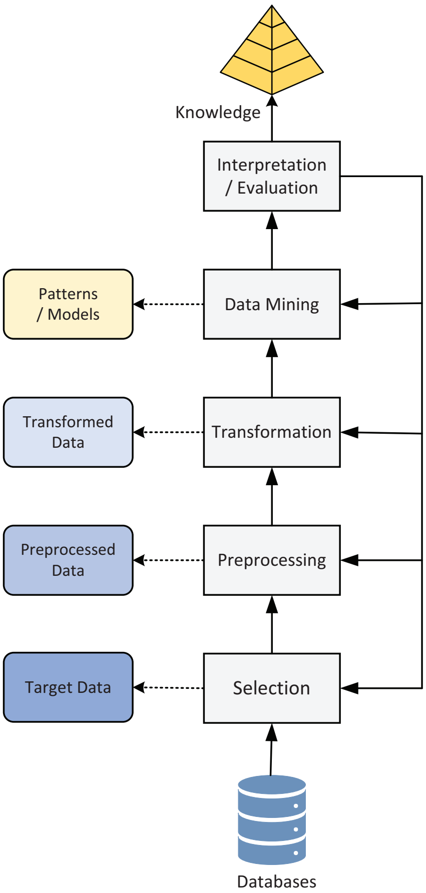


Figure 2.1: KDD general processes. *Note.* Adapted from [Fayyad et al., 1996].

1. **Data Selection**: Given a large database, this process selects a subset from the database in which the user requires the knowledge discovery system to find some useful knowledge. Thus, the data selection process accepts a large database and outputs a target data, as shown in Figure 2.1. For example, given the World Wide Web, as a massive and multi-source database, in the data selection process, a set of news stories webpages might be collected for some Web mining applications.

2. **Data Pre-Processing**: This process concerns about cleaning and removing the noise from the target data. The process also handles missing and redundant data and collects relevant

information from the required fields of the target data. For the previous case of Web content mining, some elements from the news webpages need to be removed, such as metadata, pictures, hyperlinks, tags and CS codes. Also, stop-words are usually removed, and word-stemming is performed, if required, in this process.

3. **Data Transformation**: Following the pre-processing task, the transformation process takes place. Depends on the data mining application, the pre-processed data are transformed into the required format, or some relevant features are selected to reduce the dimensionality of the pre-processed data. The selected features are then used to represent the data and used in the mining task. The output of the transformation process (i.e., the transformed data) is analogous to the subset of facts F_s discussed above, which can be passed to the data mining process.

4. **Data Mining**: This process performs a specific mining task (e.g., summarisation, classification, regression, clustering or association rule mining etc.) by searching for some interesting patterns in the transformed data. These interesting patterns can be in the form of ordered co-occurring features, a set of maximum features or even as simple as pairs of features found in the data. The effectiveness of this process can be enhanced if the user manages to perform the previous three KDD process more accurately.

5. **Results Evaluation**: The main task of the evaluation process is to ensure that the discovered patterns meet the definition of knowledge noted previously. Thus, the evaluation process must confirm that the discovered patterns are novel, valid and reflect the user's information needs. Only those patterns satisfy these criteria are considered useful knowledge.

2.1.2 Text Knowledge Discovery

Textual data have witnessed a dramatic increase since the introduction of the web in the early nineties. The amount of text continued to increase in an exponential rate, especially with the wide-spread use of social networks applications [Gao, 2015, Khan et al., 2010, Sebastiani, 2002, Wu, 2007]. This text deluge has made the search for relevant information extremely challenging as text tends to suffer from the high-dimensionality problem [Aphinyanaphongs et al., 2014, Dasgupta et al., 2007, Yang and Pedersen, 1997]. Also, textual data are sparse and noisy [Albathan et al., 2014, Algarni, 2011, Li et al., 2015]. Text-based applications also suffer from semantics related problems like ambiguity, synonymy and polysemy [Algarni and Li, 2013, Bashar et al., 2014, Li et al., 2010]. Thus, unlike other types of data, texts need special processing and analysis techniques to discover useful information from it. Knowledge discovery in text (aka text mining or text analytics) [Wu, 2007] is the process of finding meaningful and interesting patterns from text collection using different analysis tools and algorithms to suit what the user needs. Unlike knowledge discovery from databases where the stored data are usually structured (e.g., relational tables), text mining usually handles semi-structured and unstructured text, which is no more than a sequence of words or even characters [Gao, 2015, Sebastiani, 2002]. Text mining also is interdisciplinary that can span different research communities, including ML, IR, NLP and IF [Gao et al., 2015, Khan et al., 2010, Li et al., 2015, Moschitti and Basili, 2004, Zhong et al., 2012].

2.1.2.1 Text Pre-Processing

As noted above, textual data is noisy and can have an enormous number of errors and irregularities as well as noninformative words. Thus, before it can be further analysed, texts need to be pre-processed through the undertaking of some popular pre-processing tasks, including tokenisation, lemmatisation, stemming and filtering. These tasks are described below.

1. **Tokenisation**: In the tokenisation process, the text is divided into fragments of words. At the same time, punctuation is removed, and tabs (including non-text characters) are replaced with single spaces [Khan et al., 2010].

2. **Lemmatisation**: In the lemmatisation process, all verbs are converted back to the original dictionary keywords and mapped to their original infinitival forms.

3. **Stemming**: In stemming, ends of words are sliced off in anticipation of bringing them back to their dictionary keywords. For example, present participles' *–ing* endings are removed, and plurals are turned into singulars [Khan et al., 2010, Porter, 1980].

4. **Filtering**: Stop words are common words like conjunctions, articles and prepositions that are found in every text but have little or no meaning in relation to the content [Khan et al., 2010, Porter, 1980]. These are removed.

2.1.3 User Relevance Feedback

Relevance is a fundamental concept in both IR and IF. IF is mainly concerned with the document's relevance to a query about a specific subject [Li et al., 2015, 2010]. However, IF discusses the document's relevance to the user's information needs [Gao et al., 2015, Li et al., 2010]. Relevance feedback is a technique that has been extensively used mainly in IR, where a user is involved in judging the relevance of the retrieved results [Algarni, 2011, Rocchio, 1971]. A user submits a query Q to a search engine that retrieves a list of documents R, which are ranked based on their similarities to the user's query. The user, then, is involved in assessing the relevance of a top-k documents collection D to what he/she needs to either relevant (1 positive) or irrelevant (0/-1 negative) documents [Algarni, 2011]. In this case, the collection D is known as the relevance feedback such that $D \subset R$ and D has a subset of relevant documents D^+ and another subset that are irrelevant D^-. Due to the user judgement, relevance feedback has been extensively used in TM, ML, IR and IF for a variety of applications [Bashar et al., 2016, Gao et al., 2015, Li et al., 2015, Rocchio, 1971] to learn or mine useful information and knowledge.

- **Positive Feedback**: This feedback refers to the subset of documents that are relevant to what the user needs, which commonly represented as D^+ such that $D^+ \subset D$. This subset can be used to identify the main interests of the user, which makes D^+ receives much attention from many research communities [Alharbi et al., 2017b, Bashar and Li, 2018, Gao et al., 2017].

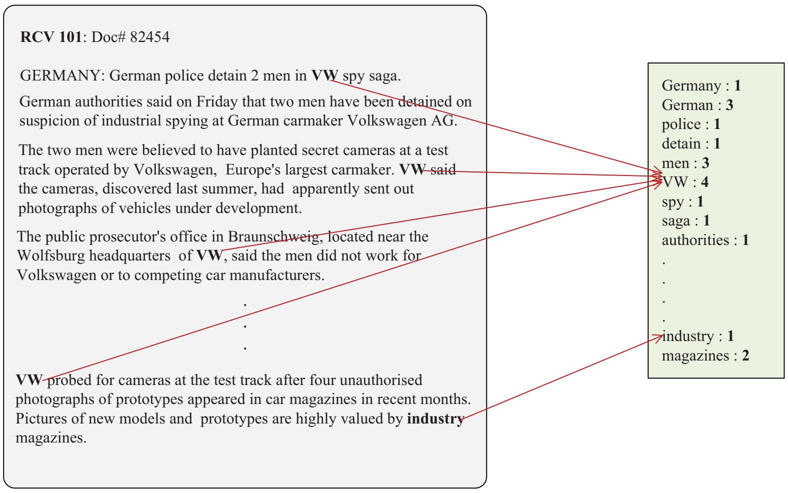
- **Negative Feedback**: The negative feedback is the subset of irrelevant documents in D, which is commonly referred to as D^- [Alharbi et al., 2018a, Li et al., 2010, 2012]. People assume that D^- documents are useful for deciding the information or knowledge that the user is not interested in. However, this assumption can be simple, knowing that these documents can be very topically diverse, which makes the identification of what exactly the user not interested in is rather challenging [Li et al., 2015, 2017c].

2.1.4 Text Representation

Text representation is a central problem in TM and ML in which a set of documents is numerically represented in a specific space [Man et al., 2009, Sebastiani, 2002, Zhong et al., 2012]. For example, given a numerical space S, and a document collection $D = \{d_1, d_2, d_3, \dots, d_m\}$ where d_x denotes the x^{th} document in D, the text representation model aims to represent each document $d_x \in D$ as a point s_x in the space S. The representation model allows the documents to be mathematically defined, as pairs of points in the space, which can be efficiently manipulated (e.g., measures the similarity or distance between two pairs) [Salton and Buckley, 1988, Salton et al., 1975]. Further, selecting the suitable representation model is crucial for the success of the TM task being undertaken [Algarni, 2011, Sebastiani, 2002]. The two primary text representation models are the keyword-based and phrase-based, as described below.

2.1.4.1 Keyword-based Representation

Representation based on keywords, the so-called bag-of-words (BoW) process (see Figure 2.2), is extensively employed in IR [Croft, 2000, Fang et al., 2004, Huston and Croft, 2014]; it is also known as the vector space model (VSM). Gerard Salton developed this model in 1960 to index and retrieve information [Salton and Buckley, 1988, Salton et al., 1975]. Most IR systems and text extraction methods have employed this model, which finds similarities among the text representations identified [Manning et al., 2008b, Rocchio, 1971]. Each document d is represented by the VSM as a vector in the feature space, $w(d) = \{x(d, t_1), x(d, t_2), \dots, x(d, t_n)\}$; the frequency of the term t is represented by each element of the vector in the document [Salton and Buckley, 1988, Salton et al., 1975].



RCV 101: Doc# 82454

GERMANY: German police detain 2 men in **VW** spy saga.

German authorities said on Friday that two men have been detained on suspicion of industrial spying at German carmaker Volkswagen AG.

The two men were believed to have planted secret cameras at a test track operated by Volkswagen, Europe's largest carmaker. **VW** said the cameras, discovered last summer, had apparently sent out photographs of vehicles under development.

The public prosecutor's office in Braunschweig, located near the Wolfsburg headquarters of **VW**, said the men did not work for Volkswagen or to competing car manufacturers.

.
.
.

VW probed for cameras at the test track after four unauthorised photographs of prototypes appeared in car magazines in recent months. Pictures of new models and prototypes are highly valued by **industry** magazines.

Germany : **1**
German : **3**
police : **1**
detain : **1**
men : **3**
VW : **4**
spy : **1**
saga : **1**
authorities : **1**
.
.
.
.
industry : **1**
magazines : **2**

Figure 2.2: The BoW representation. *Note.* Adapted from [Joachims, 1996].

Although quite useful, the VSM representation is not without limitations: It poorly represents long documents, since these contain loose values for similarity, and keywords being searched must concisely relate to the terms in the documents [Croft, 2000, Turney and Pantel, 2010]. Besides, VSM does not retain semantic information about terms [Zhong et al., 2012].

2.1.4.2 Phrase-based Representation

The keyword-based representation, resembled by the VSM or BoW, was not limitations free. The absence of semantic knowledge (e.g., words order) among keywords in VSM made it suffer from the synonymy and polysemy problems [Deerwester et al., 1990, Li et al., 2015, Luo et al., 2011]. Also, VSM considers each keyword as a separate dimension in the numerical space can be inefficient. Further, using VSM to find relations between words in documents is distance-based. It largely depends on the spatial information of vectors that represent the documents in which the sequence of the words is not considered. Thus, in an attempt to address VSM's limitations, a phrase-based representation was developed in which unstructured documents are represented by a set of phrases instead of individual keywords [Albathan et al., 2013, Fürnkranz, 1998, Huston and Croft, 2014]. A phrase is a set of keywords that appear together and carry a specific meaning.

The phrase-based representation aims to add semantic information to a word in documents by capturing its correlation with other words across the containing corpus [Albathan et al., 2013, Huston and Croft, 2014, Wang et al., 2012]. The statistical n-grams model [Manning et al., 2008b] is a widely used phrase-based representation with numerous applications in IR, IF and other related TM tasks. The n-Gram is employed to locate all the series of words that do not exceed the length of n [Albathan et al., 2013, Wang et al., 2012]. The n-gram-based text representation offers a more rigorous means of handling documents even in the presence of typographical and grammatical errors and mistakes. Besides, this model does not require any processes like tokenisation or stemming. Yet, there are also a few limitations in this model; for example, the word patterns mined with this model will be limited to a total of n, possibly limiting the discovery of long phrases or patterns [Wu, 2007].

2.2 Text Feature Selection

Having highly informative text features is crucial for the success of any text analysis application [Algarni, 2011, Li et al., 2015, Zhong et al., 2012]. Text data are extremely sparse and suffer from the high-dimensionality problem, which can cause a learning algorithm to overfit [Dasgupta et al., 2007, Khan et al., 2010, Yang and Pedersen, 1997]. Also, text documents suffer from feature redundancy and noise, which can cause knowledge discovery algorithms to be inefficient. Therefore, different TFS models and frameworks have been extensively used in the areas of TM and ML to overcome the curse of dimensionality and extract high-quality text features to support the different knowledge discovery applications [Li et al., 2015, 2017c, Tang et al., 2016, Zhang et al., 2016, Zheng et al., 2004]. Most TFS techniques exploit the statistical information (frequency) of terms and patterns in a document or set of documents for capturing the importance of the different features [Man et al., 2009, Robertson and Zaragoza, 2009, Yang and Pedersen, 1997]. However, two terms may have the same frequency in the same document, but it is hard to discover the one that contributes more semantically to the sentence. Further, the resulting features (terms or patterns) still suffer from noise and redundancy [Albathan et al., 2013, 2014, Li et al., 2015].

2.2.1 Definition

Instead of creating new features, a TFS model automatically selects a subset of features from the original set of features that discovered from a collection of documents [Combarro et al., 2005, Dasgupta et al., 2007, Forman, 2003]. The selected features must be relevant to the topics discussed in the collection that might describe the user's information needs [Gao et al., 2015, Li et al., 2015, Zhong et al., 2012]. The features must also be sufficient to represent the documents of the collection without losing important information. Additionally, the features must be meaningful and easy to understand by users and must not be redundant or noisy [Bashar et al., 2014, Wu et al., 2006]. Further, and based on the target application, the selected features have to be informative (descriptive), in the case of unsupervised TM or ML applications [Cai et al., 2010, Huston and Croft, 2014, Scott and Matwin, 1999], or discriminative (predictive or support decision making), in the case of supervised learning problems [Combarro et al., 2005, Li et al., 2010, Xue and Zhou, 2009]. By selecting such a subset of features that maintains those qualities, we can guarantee the removal of irrelevant features and, thus, reduce the total dimensionality of the feature space that a document can be mapped to.

Given a data collection, a general feature selection method selects important features from the full features set of the collection based on four main steps [Li et al., 2017b, Liu and Yu, 2005]. The steps are shown in Figure 2.3 and they are subset selection (or generation), subset evaluation, the stopping criterion and subset validation. In the first step, the subset selection searches for a subset of candidate features from the original feature set (e.g., the set of all terms in a document collection). Next, the subset evaluation step tests, based on some criteria, the goodness of the candidate subset of the first step. Before the result validation step takes place, the feature selection process must stop based on the stopping criterion. Usually, the selection process stops when the search for better features is completed, or a minimum number of features is already obtained. However, these are not the only stopping criteria available, and there are many others depend on the learning or mining application. Lastly, the result validation step measures the performance of the selected subset of features based on some ground truth (e.g., a prior known set of relevant features) [Liu and Yu, 2005].

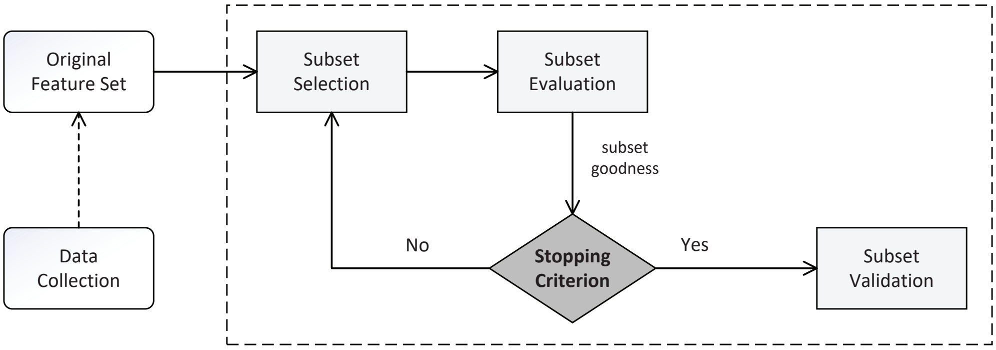


Figure 2.3: The general procedure of feature selection. *Note.* Adapted from [Liu and Yu, 2005].

2.2.2 Benefits and Challenges

A TFS model does not construct new features (i.e., features that do not belong to the original feature set that is discovered from the training samples). Instead, the model selects relevant features, removes those that are irrelevant and discards the redundant ones [Albathan et al., 2013, Algarni and Li, 2013, Li et al., 2015]. As an automatic process, TFS comes with many important benefits for learning algorithms, computer storage, decision making, and computational time [Li et al., 2017a, Liu and Yu, 2005] as follows:

1. **Improving the learning algorithm performance**: The number of features employed determine the complexity of any learning algorithm, and so, in the training set, the elimination of noisy or redundant features should enhance the accuracy of the system and improve efficiency by decreasing the process of computation [Aphinyanaphongs et al., 2014, Cai et al., 2010];

2. **Reducing data size**: Storing or retrieving features requires storage space, which can be a challenge. As it is not necessary to retrieve and store an irrelevant feature, reducing the number of features will assist in retrieving data;

3. **Enhancing data visualisation readability**: Reducing to fewer dimensions helps in providing the increased readability of the data as it also enables better visualisation and

understanding of the data [Chaney and Blei, 2012]; and

4. **Improving computational resources utilisation**: The training and testing time is also reduced when redundant and noisy features are reduced, and this can facilitate the conservation of essential computation resources like memory.

Although a TFS model can offer significant benefits to different mining and learning algorithms, the model is still subject to some challenges. For instance, choosing the most relevant feature is sensitive to the effectiveness of the selection algorithm, which indicates that there might be a chance for some important features to be missed [Alharbi et al., 2017b, Li et al., 2011]. A possible solution to prevent losing important features from the training samples is to find optimal TFS models that employ different selection criteria and integrate between them [Gao et al., 2014b, 2015]. Moreover, identifying noisy features is still challenging to most TFS models to date, and require user's involvement, which can impose further restrictions concerning the model's learning time and scalability.

2.2.3 Models Taxonomy

There is a large number of TFS models and frameworks in the current literature. To study them effectively, they have been categorised based on different characteristics. In this study, we proposed the taxonomy shown in Figure 2.4 as an attempt to study these TFS models and techniques comprehensively. Each category is separately discussed in the subsequent sections except the '*fusion*' category, which is described in Chapter 3.

One of the most widely used categorisations of TFS models is based on the search strategy employed to locate relevant features [Bolón-Canedo et al., 2013, Liu et al., 2005, Liu and Yu, 2005]. Common strategies are the filter, wrapper, embedded and hybrid strategies, as illustrated in the figure. Another categorisation approach is based on the presence and absence of semantic information in the extracted text features [Li et al., 2015, 2010]. TFS models that adopt low-level features, such as individual terms, do not consider any semantic information. However, the models that use high-level features, such as phrases, patterns, topics, concepts or a combination of them, are semantic-aware [Algarni, 2014, Bashar et al., 2017, Gao et al., 2015, Zhong et al., 2012].

Additionally, the class label information is also used to categorise different TFS models and

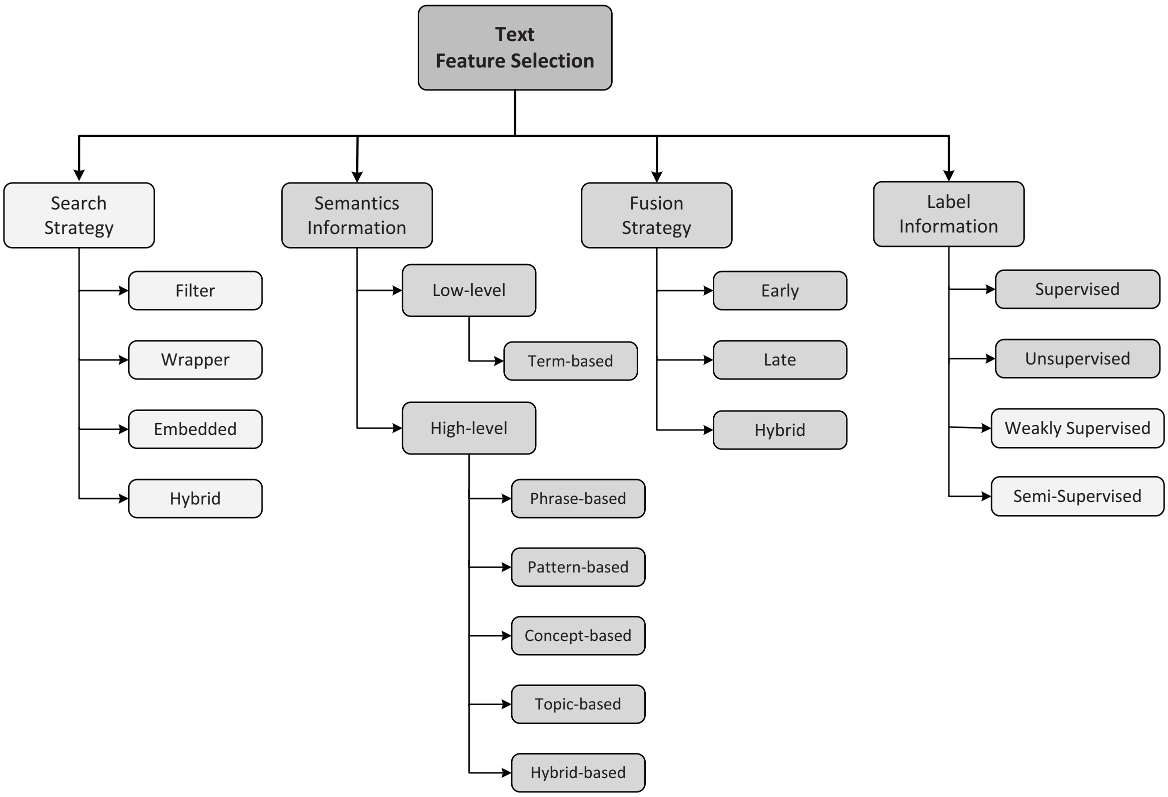


Figure 2.4: The proposed TFS taxonomy.

frameworks [Aphinyanaphongs et al., 2014, Li et al., 2017a,b]. A TFS model that requires labelled training set is called supervised while a model that does not consider the class information (i.e., use unlabelled training set) is known as unsupervised [Cai et al., 2010, Hou et al., 2010]. However, those models use a few samples of labelled data, and large samples of unlabelled data are called semi-supervised [Li et al., 2017a,b]. Weakly supervised TFS techniques do not require high-quality, human-labelled data samples for training. They can work with noisy or weakly labelled data that can be produced by learning algorithms [Baltrušaitis et al., 2019].

2.3 Search Strategy-based Models

The ultimate goal of any TFS model is selecting a subset of features from the original feature space. The selected subset is supposed to comprise the most important features (most informative or discriminative features) that represent the entire feature space almost equally [Peng et al., 2005, Song et al., 2013]. The selection or the search strategy for relevant features is one of the keys for differentiating feature selection models [Bolón-Canedo et al., 2013, Li et al., 2017a,b]. Therefore, from the search strategy perspective, a feature selection model can be classified as a filter-, wrapper-, hybrid- or embedded-based model [Bolón-Canedo et al., 2013, Liu et al., 2005]. More details about each strategy are given in the next four subsections.

2.3.1 Filter-based Models

A filter TFS model attempts to exploit the characteristics of training data samples and, then, select important features without relying on any learning algorithms (e.g., classifiers) [Liu et al., 2005, Liu and Yu, 2005]. There are two essential steps in the filter model, which are as follows: 1) based on the model's evaluation criterion, the original features are ranked; and 2) the top-ranked k features are selected [Bolón-Canedo et al., 2013, Liu and Yu, 2005]. Figure 2.5 shows the structure of the filter-based TFS model in which the feature subset selection module is the core of this model. The figure does not only show the training phase where the model selects the most important features, but it also illustrates the testing phase in which the quality of the selected features is evaluated using a learning algorithm, testing data samples and evaluation measures.

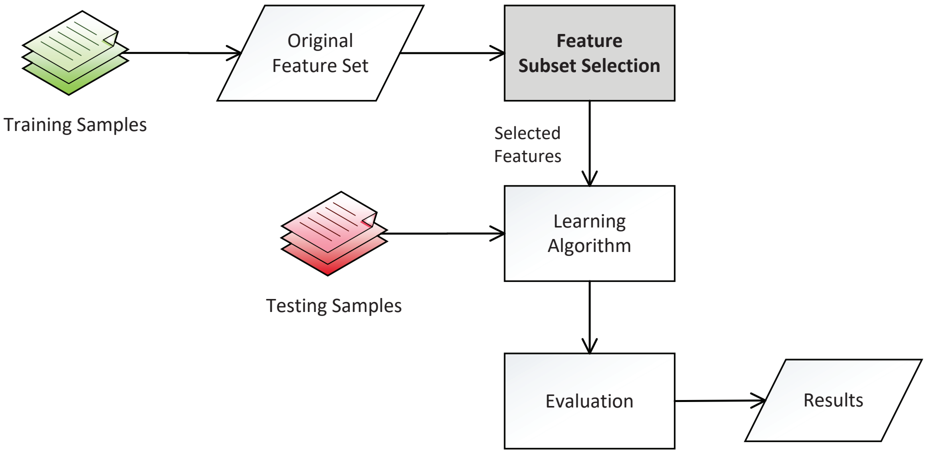


Figure 2.5: The filter model. *Note.* Adapted from [John et al., 1994]

Popular examples of TFS models that adopt the filter's strategy are information gain (IG) [Yang and Pedersen, 1997], BM25 [Robertson and Zaragoza, 2009], MI [Manning et al., 2008b], χ^2 [Chen and Chen, 2011], Prob [Jones et al., 2000a,b], TFIDF [Salton and Buckley, 1988] and other term weighting algorithms [Man et al., 2009, Wu and Gu, 2017]. The filter model is computationally efficient and can remove noisy features and simplify training data [Bolón-Canedo et al., 2013, Liu and Yu, 2005]. The model can improve the performance of any TM or ML algorithms because it does not require any classifiers to select features (i.e., not biased towards any learning algorithms) [Liu et al., 2005, Liu and Yu, 2005]. However, the filter model can miss some discriminative features and cannot validate the selected subset of features as it ignores the learning algorithm during the training phase.

2.3.2 Wrapper-based Models

A wrapper TFS model does not only exploit the characteristics of training data, as in the filter model but also uses a learning algorithm (e.g., a classifier) to assess the usefulness of the initially selected subset of features [Bolón-Canedo et al., 2013, Kohavi and John, 1997, Liu et al., 2005, Liu and Yu, 2005]. Figure 2.6 [Kohavi and John, 1997] shows the structure of the wrapper-based model and how it can be used during both training and testing phases. The shaded box in the figure represents the central part of any wrapper-based TFS model, and the learning algorithm is what distinguishes the wrapper's approach from the filter's. To select the best subset of features, the wrapper model performs three steps. First, it selects an initial subset of features from the original set based on some searching criteria. Then, the model uses the learning algorithm, as a black box, to evaluate the goodness of the selected subset of features. Lastly, the model repeats the previous two steps until the best subset of features is selected [Kohavi and John, 1997].

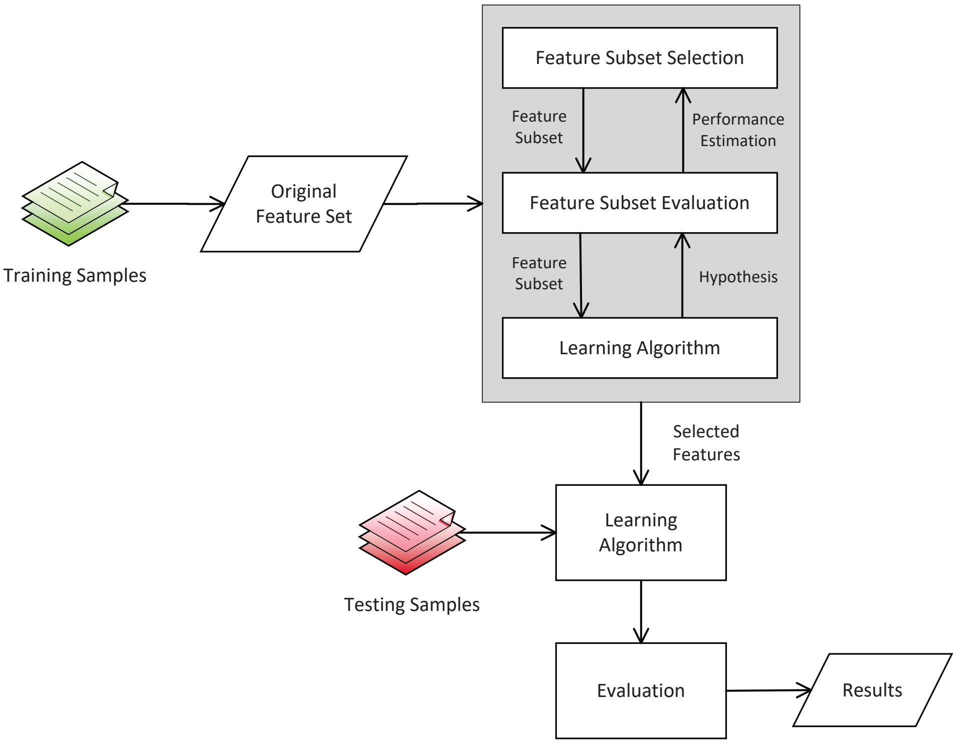


Figure 2.6: The wrapper model. *Note.* Adapted from [Kohavi and John, 1997]

Different learning algorithms, such as C4.5, naïve Bayes and ID3 [Khan et al., 2010], were employed by various wrapper-based models to validate the quality of the selected features during training [Bolón-Canedo et al., 2013, Liu and Yu, 2005]. Adopting such algorithms make the wrapper strategy more effective than the filter approach in selecting better features that might increase the accuracy of classifiers during the testing phase [Bolón-Canedo et al., 2013, Li et al., 2017b]. However, the effectiveness comes with higher computational time. Further, the wrapper model may not be applied with massive training data because it cannot manage and scale its size [Li et al., 2017a]. Also, it is well understood that the accuracy of learning algorithms during the testing phase cannot be guaranteed based on the accuracy that is estimated during the training phase. Thus, it cannot be generalised that the best subset of features selected by the wrapper model during training can perform best during testing.

2.3.3 Hybrid-based Models

The hybrid TFS model takes the advantages of both the filter and wrapper models [Liu and Yu, 2005]. As a middle-ground solution, the hybrid algorithm uses the filter model to select optimal feature subsets [Song et al., 2013]. It also exploits the learning algorithm in the wrapper model to decide the final best subset of features [Liu and Yu, 2005]. Figure 2.7 depicts the structure of the hybrid model and how the filter and wrapper algorithms are integrated. The figure also shows both the training and testing phases of the hybrid model.

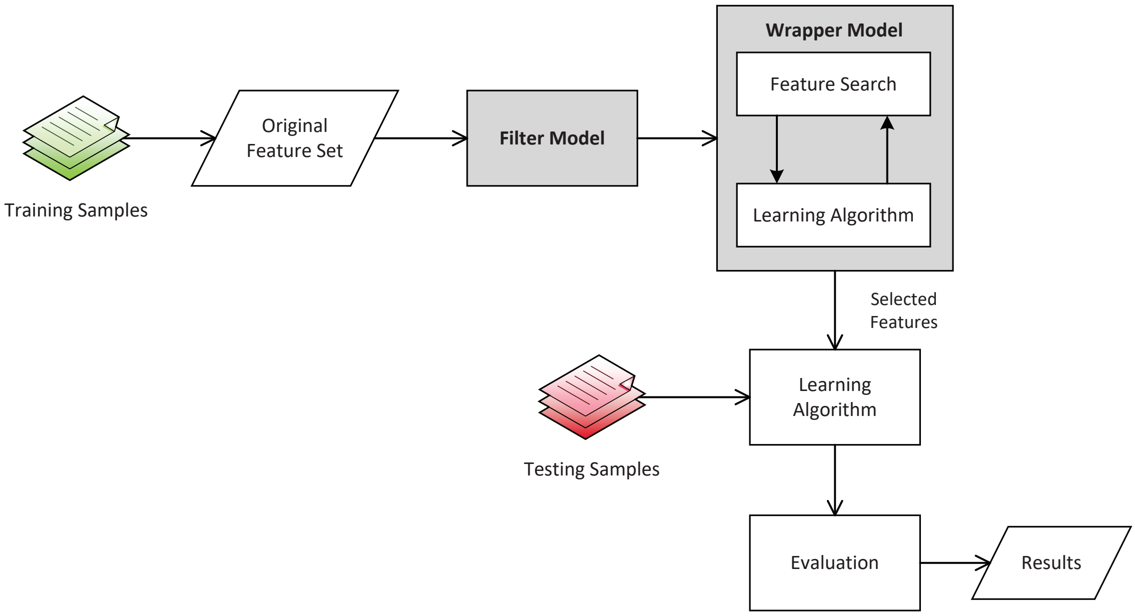


Figure 2.7: The hybrid model. *Note.* Adapted from [Albathan, 2015]

The computation of the hybrid algorithm is fast and less cumbersome, and its interaction with the learning algorithm enables it to generate an optimal set of features [Liu and Yu, 2005]. The hybrid model also can handle large data collection and does not need stopping criteria as the learning algorithm naturally performs such a task [Song et al., 2013]. However, it is challenging to guarantee the generalisability of the final subset of features selected by the hybrid model because there might be some data samples in the testing phase that could not be seen in the model training phase.

2.3.4 Embedded-based Models

Unlike the hybrid selection strategy in which the advantages of both filter and wrapper are combined, a TFS model that adopts the embedded strategy does not select features before learning a classifier [Bolón-Canedo et al., 2013]. It embeds the selection process inside the process of learning a classifier using some forms of regularisation or pruning for features [Li et al., 2017a]. Figure 2.8 illustrates the typical structure of an embedded TFS model and how the model can be used during both the training and testing phases. Based on these criteria, popular classifiers, such as random forests, weighted naive Bayes and decision tree C4.5 [Khan et al., 2010] can be considered as embedded models. Additionally, the features selected based on the SVM's weighted vectors [Joachims, 2002] and LASSO model [Tibshirani, 1996] are also regarded as embedded models [Bolón-Canedo et al., 2013].

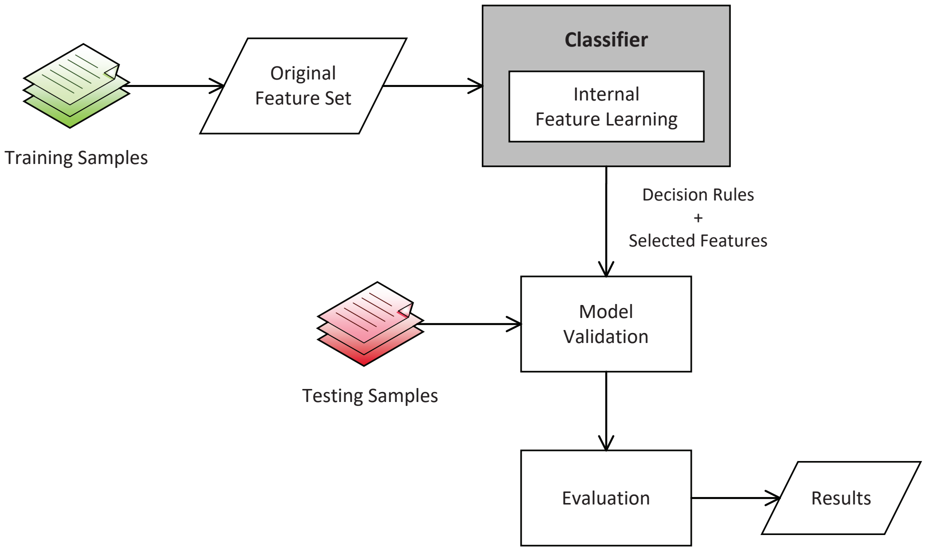


Figure 2.8: The embedded model. *Note.* Adapted from [Bolón-Canedo et al., 2013]

In the literature, many hybrid models have been categorised as embedded stressing that there is no difference between the two [Li et al., 2017b, Liu and Yu, 2005]. However, in the embedded algorithm, no feature subsets are selected before learning a classifier, as shown in the figure. Instead, the classifier can only select important features during its learning phase. The embedded model is more efficient than the hybrid one because there is no need to re-train the adopted classifier for each subset of features [Bolón-Canedo et al., 2013]. However, both models produce better results compared to the filter and wrapper algorithms [Bolón-Canedo et al., 2013, Li et al., 2017a].

2.4 Semantic Information-based Models

Based on the availability of semantic information in text features, they can be classified to low- and high-level features [Bashar et al., 2014, Li et al., 2015, 2010]. The low-levelness here implies the absence of semantic information in the features used by a TFS model. The high-level features retain some semantic information that can make them more meaningful and understood by users. More details are given in the subsequent sections.

2.4.1 Types of Text Features

In a document, text features can take different forms, such as words (terms), phrases, patterns, n-grams structures, and part-of-speech constructs (e.g., verbs, adverbs, nouns and adjectives) [Li et al., 2015, 2010]. All these terminologies refer to the *physical* features (whether lexical or syntactic) that characterise the document's text. These features can be used in representing or indexing the document for a text analysis technique. However, in the literature, the term ”feature” can also refer to some statistical attributes that are pertinent to a specific lexical or syntactic features (e.g., frequency, conditional probability distribution, etc.) [Xue and Zhou, 2009]. Li, Algarni and Zhong (2010) [Li et al., 2010] categorised text features into two groups based on the semantic information they carry in relevant documents. The first group, called high-level features, is represented by text patterns, while the second group, relates to low-level features like words. However, patterns are not the only type that belongs to the high-level group. Other features like phrases, concepts, topics or different combinations of them can also be classified as high-level features [Albathan et al., 2013, Alharbi et al., 2017c, Bashar et al., 2016, Gao et al., 2015].

2.4.2 Low-level Features

From the semantic information perspective, low-level features reside at the bottom of the semantic taxonomy of text features with almost no semantic information [Li et al., 2015, 2010]. Low-level terms (i.e., individual words) are the typical example of such features that are extensively used in IR and ML algorithms mainly to represent documents as BoW in the VSM [Albathan et al., 2013, Gao et al., 2014b]. Due to the richness of their statistical properties, low-level words were efficiently adopted by many TFS techniques in which the term's statistics were mathematically and heuristically modelled in the forms of weighting schemes [Gao et al., 2014b, Li et al., 2011, Zhou et al., 2008]. However, individual words suffer from the problem of synonymy and polysemy due to the absence of semantic relations between them [Algarni, 2014, Li et al., 2015, Zhong et al., 2012]. These problems are the leading cause of information mismatch and overload that affect many IR and IF systems [Li et al., 2008, 2012]. Thus, it is challenging to discover relevant terms that reflect the user's information needs. Some popular examples of term-based TFS models are described in the next section.

2.4.2.1 Term-based Models

- **TF*IDF**

  Term Frequency-Inverse Document Frequency [Salton and Buckley, 1988] is a commonly used weighting method in IR, TM and ML algorithms [Man et al., 2009, Sebastiani, 2002, Zhong et al., 2012]. TF-IDF linearly combines term frequency (TF) of term t in document d with the term inverse document frequency (IDF) at the collection level as follows:

  $$tfidf_{t,d} = tf_{t,d} \times idf_t \tag{2.1}$$

  where $tf_{t,d}$ is the term's t instances of occurrences in the document d.

  The term inverse document frequency is idf_t, which is employed to quantify the term's specificity in the collection of documents on the basis that terms that frequently appear in many documents are not robust determiners of specificity and should be assigned less weight compared to the ones appearing in just a small number of documents. Thus, IDF of a term t can be calculated as follows:

  $$idf_t = \log \frac{N}{df_t} \tag{2.2}$$

  where N represents the number of documents in a collection.

- **Rocchio's Algorithm**

  Rocchio's algorithm was introduced in 1971 with the SMART retrieval system [Rocchio, 1971]. Even since, the algorithm has been used extensively in IR, IF and TM applications [Li et al., 2015, 2011, Robertson and Soboroff, 2002]. Rocchio's algorithm is a relevance feedback model that uses both positive and negative documents. The algorithm uses terms to represent the documents in the VSMs in which $tf.idf$ is used to give weight to documents terms. Rocchio's algorithm can be formulated as follows:

  $$\vec{c} = \alpha \frac{1}{|D^+|} \sum_{\vec{d} \in D^+} \frac{\vec{d}}{||\vec{d}||} - \beta \frac{1}{|D^-|} \sum_{\vec{d} \in D^-} \frac{\vec{d}}{||\vec{d}||} \tag{2.3}$$

  Despite its efficiency in discriminating between positive and negative documents in the vector space, Rocchio's algorithm has low classification accuracy [Shehata et al., 2007, Yuefeng and Ning, 2006]. It is because the algorithm uses low-level terms that made it unable to cope with the problems of synonymy and polysemy.

- **Okapi BM25**

  The Okapi BM25 [Robertson and Zaragoza, 2009] is considered one of the best ranking algorithm in IR. Instead of representing documents in the VSM, BM25 uses the term t frequency and length of documents to probabilistically assign a weight to the term at the collection level using the following equation:

$$w(t) = \frac{tf \times (k_1 + 1)}{k_1 \times \left((1-b) + b\frac{DL}{\text{AVDL}}\right) + tf} \times \log \frac{\frac{(r+0.5)}{(n-r+0.5)}}{\frac{(R-r+0.5)}{(N-n-R+r+0.5)}} \tag{2.4}$$

  where N is the number of training documents; R is the number of positive documents in the training set; tf is the term frequency; b and k_1 are experimental parameters whose values set to 0.75 and 1.2 respectively as recommended in [Manning et al., 2008b]; DL and $AVDL$ are the document length and the average document length; n and r are the total number of documents contain the term t, and the total number of positive documents that include the same term t.

- **Mutual Information**

  The mutual information (MI) [Manning et al., 2008b] is derived from information theory and widely used for measuring the mutual dependency between terms given a specific collection or class label. Thus, given a collection of documents that represent a particular topic of interest and a term t, the mutual information can be calculated as follows:

$$mi(t) = \log \frac{r \div R}{n \div N} = \log \frac{r}{R} - \log \frac{n}{N} \tag{2.5}$$

  where R, r, N and n denote the same statistical information of BM25 as described above.

- **Chi-Square**

  The chi-square (χ^2) [Chen and Chen, 2011] is a statistical test that is widely used to measure the strength of independence between a term t and a specific topic (i.e., a collection of documents). Chi-square's equation can be written as follows:

$$\chi^2(t) = \frac{N \times (r \times N - n \times R)^2}{R \times n \times (N - R) \times (N - n)} \tag{2.6}$$

where R, r, N and n denote the same statistical information of BM25 and MI as described above.

- **Probabilistic Models**

  Four term-based methods are proposed by Jones et al. [Jones et al., 2000a,b] and used as relevance ranking functions for retrieval models. These probabilistic methods assign weights to search terms based on the independence and ordering assumptions for binary relevance models. The four weighting functions are as follows:

$$F_1(t) = \log \frac{(r \div R)}{(n \div N)} \tag{2.7}$$

$$F_2(t) = \log \frac{(\frac{r}{R})}{(\frac{n-r}{N-R})} \tag{2.8}$$

$$F_3(t) = \log \frac{(\frac{r}{R-r})}{(\frac{n}{N-n})} \tag{2.9}$$

$$F_4(t) = \log \frac{(\frac{r}{R-r})}{(\frac{n-r}{N-n-R+r})} \tag{2.10}$$

  where t is an individual term, r is the number of documents in D^+ that contain the term t, n is the number of documents in D that contain t, N and R denote the total number of documents in D and D^+, respectively. Based on the experiment in [Zhong et al., 2012], which was conducted on the RCV1 dataset, the following function performed best compared to the others above:

$$W(t) = \log \left(\frac{r + 0.5}{R - r + 0.5} \div \frac{n - r + 0.5}{(N - n) - (R - r) + 0.5} \right) \tag{2.11}$$

- **LASSO**

  LASSO [Tibshirani, 1996] is a linear regression model and stands for Least Absolute Shrinkage and Selection Operator. LASSO is considered as an embedded TFS model as it uses l_1-norm regularisation to eliminate unimportant features by forcing their weights ($\bar{w}$) to be zero though some optimisation methods. Once $\bar{w}$ is calculated, the features are sorted in descending order, and top-k can be selected. LASSO was used in this study is the same way as in [Li et al., 2015].

Despite the efficiency of the term-based models described above, they suffer from the limitations of low-level terms. These models cannot handle the synonymy and polysemy problems

[Li et al., 2015, 2012]. The models do not assume that documents can discuss multiple topics [Alharbi et al., 2017c, Gao et al., 2015]. Additionally, term-based TFS models are sensitive to the noisy terms in the collection and cannot manage the uncertainties in relevant documents [Albathan et al., 2013, 2014]. To overcome the weaknesses of low-level terms, many high-level features were proposed, as described in the following section.

2.4.3 High-level Features

Some text features are attributed as high-level due to the semantic information they contain [Li et al., 2015, 2012]. Popular examples are phrases, patterns, topics, concepts or a mixture of these features. Phrases were extracted and probabilistically modelled to understand the semantic meaning of user information needs and, thus, improve the performance of many IR, IF and TM applications [Albathan et al., 2013, Fürnkranz, 1998, Wang et al., 2012]. Similarly, different association rule mining methods were adapted and employed to discover interesting text patterns [Li et al., 2012, Wu et al., 2006, 2004, Zhong et al., 2012]. These patterns are semantically meaningful and used understand user information preferences for many TM problems [Albathan et al., 2013, 2014, Li et al., 2015, 2010, Zhou et al., 2011]. Ontological concepts were also used to add an explicit semantic layer to user information needs and, thus, interpret their meanings for more reliable results [Bashar and Li, 2017, 2018, Egozi et al., 2008]. Statistical topic modelling algorithms were also adopted to discover the topics that user might interested in and, therefore, discover the more relevant topical features [Blei et al., 2003, Hofmann, 2001, Wei and Croft, 2006]. Additionally, these features above were also integrated to solve the limitations of specific other high-level features. Several popular and state-of-the-art TFS models that adopted high-level features are described and discussed in the following sections.

2.4.3.1 Phrase-based Models

- **n-Grams**

  The n-grams method extract a sequence of terms (words) or characters from a document by moving a sliding window of size n [Albathan et al., 2013, Fürnkranz, 1998]. The simplest form of n-grams is the unigram that can be extracted by assigning $n = 1$ (i.e., move the window one place at a time). The extracted n-grams become more meaningful and interesting with higher values for n, such as bigram ($n = 2$) and trigram ($n = 3$) etc. As phrases are more semantically rich than individual words, the n-grams language model has been widely used in

IR, IF and other TM applications [Lavrenko and Croft, 2001, Robertson and Zaragoza, 2009, Wang et al., 2007]. A common way to model a sequence of terms of n-grams is through the conditional probability of a term given the preceding term [Albathan et al., 2013]. Thus, if n-grams $= \{t_1, t_2, t_3, \ldots, t_n\}$ then it can be modelled as follows:

$$P(t_1 t_2 t_3 \ldots t_n) = P(t_1)P(t_2|t_1 t_2) \ldots P(t_n|t_1 t_2 \ldots t_{n-1}) \tag{2.12}$$

where the conditional probability of a term t_n given its preceding term t_{n-1} ($P(t_n|t_{n-1})$) can be estimated using the following equation:

$$P(t_n|t_{n-1}) = \frac{P(t_{n-1}, t_n)}{P(t_{n-1})} \tag{2.13}$$

Despite the meaningfulness of phrases (n-grams), they did not show encouraging performance in discovering relevant features that reflect user information needs, as can be seen in many studies [Gao et al., 2015, Li et al., 2015, Moschitti and Basili, 2004, Scott and Matwin, 1999, Wu et al., 2006]. One of the main reasons behind the poor performance of n-grams models is the existence of noisy terms [Albathan et al., 2013, Fürnkranz, 1998]. The strict sequential appearance of terms in n-grams made it challenging to handle noisy terms and allows them to be modelled alongside with important terms. Additionally, n-grams language models cannot manage uncertainties in relevant documents and do not assume that these documents might discuss multiple topics and themes [Alharbi et al., 2017c, Gao et al., 2015].

2.4.3.2 Pattern-based Models

Text patterns, as sets of associated terms, are widely used in different TM, IR and IF applications [Gao et al., 2015, Li et al., 2015, 2012, Wu et al., 2019, Zhong et al., 2012]. Many pattern mining algorithms are used to extract interesting text patterns, such as frequent patterns [Han et al., 2007], closed patterns [Yan et al., 2005], sequential patterns [Mooney and Roddick, 2013], maximal patterns [Feldman et al., 1997] and master patterns [Yan et al., 2005]. These different types of patterns are employed by many pattern-based TFS models to discover relevant features that describe user information needs [Algarni et al., 2010, Li et al., 2011, 2010, 2012, Wu et al., 2004]. Some state-of-the-art examples are described below.

- **MP**

  The master pattern model [Yan et al., 2005] groups frequent closed patterns into clusters (aka pattern profiles or master patterns) based on some similarity's measures. The model was developed on the basis that individual text patterns might not be representative but assembling them together in one master pattern can increase their informativeness and lead to a better discovery of knowledge. To summarise the set of closed patterns $CP = \{cp_1, cp_2, cp_3, \ldots, cp_n\}$ that was discovered from the set of all paragraphs G of relevant documents D^+, the MP model defines a master pattern M as a triple $\langle \mathrm{P}, \phi, \rho \rangle$ where P is a probability distribution vector of the pattern terms, ϕ is the set of closed patterns and ρ is the pattern support. The model also combines master patterns that are closed in the distance into a single one and uses the k-means clustering algorithm to generate the user-specified k number of master patterns.

- **PDS**

  Pattern Deploying Based on Support [Zhong et al., 2012] is one of the state-of-the-art pattern-based feature selection models that adopt the late fusion concept. It is an enhanced extension to the PTM [Wu et al., 2004] and the PDM [Wu et al., 2006] to overcome the limitations of pattern frequency and usage. PDS extracts closed sequential patterns in relevant documents as a high-level features that represent user's information needs based on a threshold of minimum support (min_sup). Then, the model deploys all the extracted patterns into terms where each term's score (also called *support*) can be calculated using the following equation:

$$\mathrm{support}(t, D^+) = \sum_{i=1}^{n} \frac{|\{p | p \in SP_i, t \in p\}|}{\sum_{p \in SP_i} |P|} \tag{2.14}$$

  where D^+ is the relevant documents in the training set and n is the total number of D^+; $|P|$ is the total number of terms in pattern p; SP is the set of all closed sequential patterns in D^+.

- **RFD**

  The relevance feature discovery model [Li et al., 2015, 2010] is one of the state-of-the-art TFS techniques that uses high-level pattern to weight low-level terms. The RFD model clusters terms into three groups—positive specific, general and negative specific— based on their appearance in the positive D^+ and negative D^- training documents. This clustering helps to determine the specificity of each individual term to represent the document collection that discuss user information needs. Given a term t, the RFD model defines its specificity using the following equation:

$$\mathrm{spe}(t) = \frac{|coverage^+(t)| - |coverage^-(t)|}{n} \tag{2.15}$$

  where $n = |D^+|$, the function $\mathrm{coverage}^+(t)$ is defined to be $\{d \in D^+ | t \in d\}$ and, inversely, the function $\text{coverage}^{-}(t)$ is as $\{d \in D^{-} | t \in d\}$.

Based on the value of $\text{spe}(t)$, for example, if the $\text{spe}(t) > 0$, then, the RFD model assumes that the term t can be more relevant to D^{+} rather than D^{-}. The model uses the classification rule $G = \{t \in T | \theta_1 \leq spe(t) \leq \theta_2\}$ to group general terms together in the set G and similarly uses the rule $T^{+} = \{t \in T | spe(t) > \theta_2\}$ for the specifically positive terms T^{+} and the rule $T^{-} = \{t \in T | spe(t) < \theta_1\}$ for the specifically negative terms T^{-}. Both θ_1 and θ_2 are experimental coefficients that denote the minimum and maximum bounds of general terms specificity, respectively.

The RFD model selects some top-K irrelevant documents (called offenders) to revise the estimated weights of terms based on the specificity function $spe(t)$ and the support $w(t)$ of the mined sets of closed sequential patterns SP that terms appear in as follows:

$$weight(t) = \begin{cases} w(t) + w(t) \times spe(t), & \text{if } t \in T^{+} \\ w(t), & \text{if } t \in G \\ w(t) - |w(t) \times spe(t)|, & \text{if } t \in T^{-} \end{cases} \tag{2.16}$$

The RFD model assumes $w(t) = w(t, D^{+})$ based on the following equation:

$$w(t, D^{+}) = \sum_{i=1}^{n} \frac{|p | p \in SP_i, t \in p|}{\sum\limits_{p \in SP_i} |p|} \tag{2.17}$$

where $n = |D^{+}|$, p is a closed sequential pattern and $|p|$ is the length of p (i.e., the number of terms in the pattern p).

- **PCM**

  The pattern co-occurrence matrix model [Albathan et al., 2012] defines a $n \times n$ matrix over a relevant document to represent the co-occurrence relationships between the patterns extracted from all paragraphs of the document collection. The matrix is used to remove the noisy patterns through a re-evaluation process. The PCM model uses a set of closed sequential patterns P extracted using a small minimum support ($min\_sup = 0.2$). Thus, given a matrix A and $P = \{p_1, p_2, p_3, \ldots, p_n\}$, the matrix element A_{ij} holds the the number of times pattern p_i comes (i.e., co-occur) after pattern p_j in the same paragraph. The PCM uses $W_R(p_i) =$

$\sum_{j}^{n} A_{i,j}$ to calculate the total co-occurrence of pattern p_i in relevant document d_i for a row in the defined matrix. Similarly, the model also uses $W_C(p_i) = \sum_{j}^{n} A_{j,i}$ for a column. Then, PCM sums the total co-occurrences of the same pattern as $PCM(p_i) = W_R(p_i) + W_C(p_i)$ before it normalises it based on the length of the target document as follows:

$$PCM(p_i) = \frac{W_R(p_i) + W_C(p_i)}{n \times m} \tag{2.18}$$

where $n = |P|$ and m is the total number of paragraphs in the relevant document.

- **SCSP**

  The specific closed sequential patterns [Albathan et al., 2014] uses the ERS $\xi : T \to 2^{P\times[0,1]}$ to weight patterns Ptn based on the term distribution in patterns and the pattern distribution in documents such that $\xi(t) = \{(ptn, f(ptn))|t \in ptn, f(ptn) > 0\}$ and $f(ptn)$ can be calculated as follows:

$$f(ptn) = \frac{\sum_{d\in D^+} supp_a(ptn, D^+)}{\sum\limits_{d\in D} supp_a(ptn, D)} \tag{2.19}$$

  where ptn is a text pattern and $supp_a(ptn, D^+)$ calculates the absolute support of ptn in D^+ paragraphs. Then, the SCSP model finds the specific closed sequential patterns based on the weight $pr(ptn)$ for all patterns $ptn \in Ptn$. The weight is estimated as follows:

$$pr(ptn) = f(ptn) \times \sum_{t\in ptn} p(t) \tag{2.20}$$

  where $p(t) = \frac{w(t)}{\sum\limits_{t_j\in T} w(t_j)}$ and $w(t) = tfidf(t)$. The set of specific closed sequential patterns represent the collection of documents.

As text patterns brought some interesting semantic information to the field of TFS for relevance discover, they also come with many challenges. First, selecting only relevant patterns out of a vast number of extracted patterns is challenging as many of these patterns are noisy, redundant and difficult to be interpreted [Bashar and Li, 2018, Gao et al., 2015, Li et al., 2015]. However, if some interesting patterns are selected, the selection process still experimental and might lead to the loss of some important patterns or terms [Alharbi et al., 2017b,c]. Also, pattern mining algorithms seem to impose further time-complexity and scalability problems when used with more massive datasets. Further, the pattern interestingness measures (e.g., support, confidence, etc.) are not informative about the relevance of patterns to the user information needs [Li et al., 2015, Zhong et al., 2012]. Additionally, pattern mining models do not assume that text documents can exhibit multiple topics and cannot handle the uncertainties in relevant long documents [Alharbi et al., 2018b, Gao et al., 2014b, 2015].

2.4.3.3 Topic-based Models

A topic, as a set of semantically related terms, has extensively used in TM and ML problems, such as classification [Soleimani and Miller, 2016], clustering [Yin and Wang, 2014], summarisation [Wu et al., 2019], retrieval [Wei and Croft, 2006], filtering [Gao et al., 2015] and many others. Given a text corpus, a topic can be statistically generated from the corpus using a probabilistic topic modelling algorithm. The most popular examples of such algorithms are PLSA [Hofmann, 2001], LDA [Blei et al., 2003] and their variations. However, LDA is widely used, as a fully generative Bayesian model, and more effective than PLSA [Blei et al., 2003, Gao et al., 2015, Wei and Croft, 2006]. Both algorithms assume that a document can discuss multiple topics, which, in many cases, reflect the natural structure of long documents. Thus, these algorithms represent the document as a mixture of topics in which each topic is a probability distribution over all terms in the corpus [Blei et al., 2003, Gao et al., 2014b]. This probabilistic representation is efficient and can reduce the corpus's dimensionality to just a limited number of topics [Bashar et al., 2016, Gao et al., 2015]. The topic representation itself allows the selection of most probabilistically relevant terms based on their distribution in each document and the entire corpus. More details about the PLSA model is given next, and the LDA is extensively discussed in the next chapter.

- **PLSA**

  Probabilistic Latent Semantic Analysis (PLSA) [Hofmann, 2001] is an enhanced probabilistic model of the LSA, which is a linear algebra-based model. The PLSA (see Figure 2.9) is a statistical topic model that relaxes the simple assumption of the unigram model that states a document d contains only one topic. PLSA represents each document as a mixture of hidden topics. However, given a latent topic z, as an unobserved variable, the PLSA model was developed on the basis that a document d and a term t are conditionally independent [Blei et al., 2003], as can be seen from Figure 2.9 where the document is a sequence of N terms from a collection of M documents. Thus, PLSA it is not entirely generative model in the sense that it cannot generate new documents (infer unseen documents during the training phase).

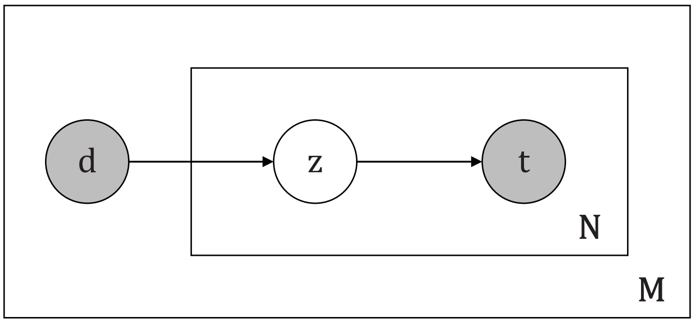


Figure 2.9: The graphical representation of the PLSA model in which d and t are the only observable variables. *Note.* Adapted from [Blei et al., 2003]

The PLSA model can deal with the problem of polysemy [Hofmann, 2001] and will be used as a baseline in our experiment in which a term t_i in relevant document d weight can be calculated as per the following equation:

$$p(d, t_i) = p(d) \sum_{j=1}^{|Z|} p(t_i|z_j) \times p(z_j|d) \tag{2.21}$$

where $|Z|$ is the total number of topics.

Apart from their mathematical soundness and the flexible representation they produce, topic-based models adopt the BoW representation in which the order of terms is ignored [Blei, 2012, Blei et al., 2003, Wei and Croft, 2006]. The models also are sensitive to term frequency, knowing that most frequent terms are general and less specific to the main topic in a document. As probabilistic models, both PLSA and LDA are biased towards frequent topics in the text corpus [Ding and Yan, 2015, Mimno et al., 2011, Xu et al., 2019], which can overshadow other equally relevant but less frequent topics. The generated topics can be difficult to be understood due to their term-based representation, and the topics, generally, lack explicit semantics [Bashar and Li, 2017, Bashar et al., 2016]. Further, as noted previously, both PLSA and LDA cannot handle uncertainties in long documents and cannot deal with negative feedback. All these issues made the PLSA and LDA models ineffective for selecting relevant terms that describe user information needs, as shown in the experimental studies of the researches in [Alharbi et al., 2017b,c, Bashar et al., 2016, Gao et al., 2014b, 2015].

2.4.3.4 Concept-based Models

Another approach to overcome the semantic limitations of the above text features is to use ontological concepts. Concepts reside at the highest level in terms of explicit human understanding to the real world [Bashar and Li, 2017, Bashar et al., 2016, Shehata et al., 2007]. A concept can be defined as a set of semantically related words that together represent a specific object or idea in domain-specific, human background knowledge [Abul Bashar, 2017, Bashar et al., 2016]. Commonly, a knowledge-base ontology of a specific domain is used to mimic such background knowledge. A domain-specific ontology is simply a set of concepts that are connected by specific semantic relations (e.g., Part-of, Is-a, Related-to, etc.) [Tao, 2009, Tao et al., 2011, Yuefeng and Ning, 2006]. Domain ontologies are widely used in information gathering and semantic Web mining applications [Abul Bashar, 2017, Tao et al., 2011]. Also, they were used to interpret and understand the meanings of text features, such as terms [Egozi et al., 2008, Shen et al., 2012b], phrases [Bing et al., 2015, Shehata et al., 2007], patterns [Bashar and Li, 2018, Bashar et al., 2014, 2017] and topics [Bashar and Li, 2017, Bashar et al., 2016], for discovering relevant features that describe user information needs. A few state-of-the-art concept-based TFS models are described below.

- **CBM**

  Concept-based model (CBM) [Shehata et al., 2007] defines a concept as a labelled term (word or phrase) that contributes to the semantics of a sentence in a document. This term can then be analysed based on its importance at two different levels, namely, the sentence and document levels. A concept can also be used as a text feature for measuring the similarity of documents, which can be used in different text mining tasks like clustering and classification. A conceptual TF (CTF) model was proposed in [Shehata et al., 2010] for measuring the similarity between documents based on the analysis of concepts at the sentence and document levels. At the sentence level, $ctf(c)$ is defined as the number of times a concept c occurs as an argument of a verb structure in a sentence s and can be normalised as $ctf_{weight}(c)$. The more c appears as an argument of different verb structures in a sentence, the more it contributes to the meaning of sentence s. That is how each concept can be analysed at the sentence level.

  In contrast, at the document level, each concept is analysed by calculating the frequency of the concept term (word or phrase) in a document and represented and normalised as $ct(c)$, and $ct_{weight}(c)$, respectively. Based on the above definitions, two different concept weights can be measured—at the sentence level and document level—for the same concept in which its weight can be calculated as $weight(c) = ctf_{weight}(c) + ct_{weight}(c)$

- **POM**

  Personalised ontology model (POM) [Shen et al., 2012b] uses the RFD model [Li et al., 2015, 2010] to discover relevant terms from both relevant and negative document sets, and then maps (i.e., annotates) these terms to the concepts of the Library of Congress Subject Headings (LCSH) ontology. The POM model assumes that relevant terms are semantically independent in each document. However, such an assumption can be too simple knowing that relevant documents might share similar topics in which many terms are semantically related. This simple assumption made POM ineffective in annotating relevant terms and, thus, representing user information needs.

- **PIM**

  Pattern Interpretation Model (PIM) [Bashar et al., 2014] attempts to interpret text patterns using high-level concepts taken from the LCSH ontology. The model mines closed patterns from relevant documents and, then, summarises them to a set of master patterns. To explain the meaning of master patterns and made them understandable to humans, PIM performs four steps. First, it estimates the concept's support based on the overlap between the terms in concepts and patterns. Then, PIM deploys and estimates the relevance weight for each of the overlapping term in matched concepts. Lastly, as it can be no overlapping between some concepts and patterns, the PIM model adds the non-overlapping terms in patterns as new concepts and estimates their relevance independently. The model achieved better results compared to the pattern-based PDM [Zhong et al., 2012] model.

Despite their explicit specification of meaning, ontological concepts are not limitation free. Domain-specific ontologies can be incomplete, imprecise, vague and difficult to be updated [Li and Zhong, 2004, Tao, 2009, Yuefeng and Ning, 2006]. Human-defined concepts, which are constructed manually, are expensive and time-consuming. Automatically discovered ones are less in terms of meaning and interpretation that make them hard to be understood [Abul Bashar, 2017, Bashar and Li, 2018]. Additionally, the use of knowledge-base ontology in TFS impose further computational time-complexity and cannot estimate the relevance of semantically related features. Ontologies also do not provide clear mechanism to identify and group similar topics that co-occur in relevant documents. All these limitations can significantly impact and limit the use of ontological concepts for identifying relevant features that represent user information preferences. The experimental results in [Bashar and Li, 2017, Bashar et al., 2016, 2014, Shen et al., 2012b, Tao et al., 2011, Yuefeng and Ning, 2006] confirm the negative effects of using ontological concepts in TFS for relevance discovery.

2.4.3.5 Hybrid-based Models

On the basis that each of the high-level features has limitations, many studies combine these features in a unified framework to exploit their advantages. For example, phrases and patterns were combined in [Albathan et al., 2013] to remove noisy phrases through the exploitation of patterns taxonomic relations. Also, topics and patterns were integrated in [Gao et al., 2017, 2014b, 2015] to understand user information needs by benefiting from the multi-topic assumption of documents in topic modelling and positional relations of text patterns. Similarly, ontological concepts were incorporated with patterns [Bashar and Li, 2018, Bashar et al., 2017] and topics [Bashar and Li, 2017, Bashar et al., 2016] to understand the meaning of these statistical and semantic structures and use them to identify user information preferences. Some state-of-the-art hybrid-based models are discussed below.

- **TNG**

  The Topical n-Grams model [Wang et al., 2007] integrates topic model with phrases (n-Grams) to discover topical phrases that are more discriminative and interpretable. Thus, TNG can be considered as another type of hybrid-based model that uses latent topic and phrase as representative text features. TNG has been treated as a relevance ranking model in our experiment as it appears in [Gao et al., 2014b] as follows:

  $$rank(d) = \sum_{j=1}^{V} \sum_{k=1}^{n_j} count(ph_{jk}) \times \vartheta_{D^+,j} \tag{2.22}$$

  where $rank(d)$ is the relevance ranking of document d to the user information needs; $count(ph_{jk})$ is the frequency of phrase k in topic j which represents the topic relevance; finally, $\vartheta_{D^+,j}$ is the proportion of topic j in relevant documents D^+; V is the number of topics; n is the number of phrases.

- **Pattern-based Topic Models**

  Pattern-based Topic Models (PBTM) [Gao et al., 2013, 2014b, 2015] are also another type of hybrid-based TFS models where topics and patterns have been incorporated to obtain semantically rich and discriminative representation for information filtering. PBTM-FP and PBTM-FCP [Gao et al., 2013] integrate frequent patterns (FP) and frequent closed patterns (FCP) with latent topics to represent user profiles. The models are also treated as relevance ranking models where the document relevance ranking can be calculated as follows:

$$rank(d) = \sum_{j=1}^{V} \sum_{k=1}^{n_j} |PA_{jk}^{d}|^{m} \times f_{jk} \times \vartheta_{D^{+},j} \tag{2.23}$$

where $rank(d)$ is the estimated relevance ranking for document d; V is the number of LDA topics; n is the number of used patterns (FP or FCP); the parameter m is the pattern specificity scale and set to 0.5; $|PA_{jk}^{d}|^{m}$ is the matched patterns for topic j in document d; the support of matched pattern is f_{jk}; finally, $\vartheta_{D^{+},j}$ is the proportion of topic j in relevant documents D^{+}. The SPBTM model [Gao et al., 2014b] enhances LDA topics by combining them with significant matched patterns (SMPatterns). The model is also treated as a relevance ranking model based on the following equation:

$$Rank_{E}(d) = \sum_{j=1}^{V} \sum_{k=1}^{n_j} \sum_{X \in SM_{jk}^{d}} \eta x |X|^{0.5} \times \delta(X, d) \times f_{jk} \times \vartheta_{D,j} \tag{2.24}$$

where SM_{jk}^{d} represents the significant matched patterns set of the equivalence class EC_{jk}, X is a matched pattern in document d, $\vartheta_{D,j}$ is the j^{th} topic distribution, f_{jk} is the statistical significance of the equivalence class and $\delta(X, d)$ is a function defined as follows:

$$\delta(X, d) = \begin{cases} 1, & \text{if } X \in d \\ 0, & \text{otherwise} \end{cases}$$

Similarly, the MPBTM model [Gao et al., 2015] is developed for IF and adheres almost to the same steps of the SPBTM model. However, MPBTM integrates the maximum matched patterns instead and estimates the document relevance as follows:

$$Rank_{E}(d) = \sum_{j=1}^{V} \sum_{k=1}^{n_j} |MC_{jk}^{d}|^{0.5} \times \delta(MC_{jk}^{d}, d) \times f_{jk} \times \vartheta_{D,j} \tag{2.25}$$

where MC_{jk}^{d} is the set of all maximum matched patterns. For all models, it is assumed that the higher the ranking value of $Rank_{E}(d)$, the more likely that document d meets user information needs.

- **LdaConcept**

  The latent Dirichlet allocation concept-based model [Chemudugunta et al., 2008] uses LDA and a knowledge base (an ontology) to label a set of documents D using a set of human-defined concepts C obtained from the used ontology. The model defines a concept as a set of unique words taken from a standard ontology. The LdaConcept model imitates the LDA and defines each concept c_j as a constrained topic in which $p(w_i|c_j) = 0$ if the word $w_i \notin c_j$. The model also similar to LDA and defines a document from a probabilistic mixture of document-specified concepts.

Thus, LdaConcept estimates the probability of the conceptual word w_i being relevant to a document d as follows:

$$p(w_i|d) = \sum_{j=1}^{|C|} p(w_i|c_j) \times p(c_j|d) \tag{2.26}$$

where $p(w_i|c_j)$ and $p(c_j|d)$ were also inferred similarly using the Gibbs sampling algorithm as in the LDA.

While integrating different high-level features comes as an approach to exploit the advantages of each type, this approach is not without limitations. In addition to the time-complexity and scalability issues that can be imposed on the intended applications, this approach is also sensitive to the restraints of the candidate features. For example, this sensitivity problem can be seen in the TNG [Wang et al., 2007] model in which phrases and topics were probabilistically modelled using Bayesian theory. Despite the sophistication of the proposed theory, TNG did not perform well as shown by many studies [Alharbi et al., 2017c, Gao et al., 2017, 2014b, 2015]. Possible reasons behind the TNG's inferior performance is noisy phrases and terms derived by the strict sequential occurring of these terms in phrases. The same problem continues to occur in the patter-based topic models [Gao et al., 2017, 2014b, 2015] where different types of patterns combined with latent topics. These models could not address the problems of noisy patterns, interestingness measures and the informativeness of some topical features. Similarly, adding explicit semantics to patterns and topics though ontological concepts [Bashar and Li, 2017, 2018, Bashar et al., 2016, 2017] could not solve the patterns or the topics problems because none of their original problems was effectively addressed, especially when uncertainties exist in relevant documents [Alharbi et al., 2018a].

2.5 Label Information-based Models

Feature transformation (or extraction) techniques, such as the principal components analysis model, can reduce the dimensionality of datasets by forming new low-level feature space [Anastasiu et al., 2013, Liu et al., 2005]. This method has been successfully used in many text mining applications, and it reduces the total dimensionality of a document collection; however, the new low-level representation (text features) does not truly represent the original document set [Anastasiu et al., 2013, Cai et al., 2010, Liu et al., 2003]. Thus, it is hard to trust such a reduction, and it is not justifiable for further text analysis. In contrast, feature selection can select only informative features from the original set of features, without requiring any further transformations [Forman, 2003, Li et al., 2017a,b, Liu et al., 2005]. This can reduce the overall dimensionality by focussing on a certain set of features that are more relevant to a text-domain analyst.

In the relevant literature, due to TFS's importance, there are various TFS techniques; however, they use different algorithms, which makes them difficult to study comparatively. Despite the multiplicity of approaches, they all process just two types of text data, namely, supervised and unsupervised data [Li et al., 2017a, Man et al., 2009, Wang et al., 2017]. Supervised data have been manually labelled by domain experts, whereas the unsupervised ones are still unlabelled (as they may naturally exist in an information repository). Therefore, broadly speaking, TFS methods can be categorised, based on the class label information, into supervised, semi-supervised, weakly supervised and unsupervised models [Li et al., 2017a]. However, the label information does not give sufficient details about the internal structure of the TFS model. Thus, in this study, each category can also be classified into filter, wrapper and hybrid (or embedded) models based on their internal learning algorithms.

Further, it is worth mentioning that some TFS techniques can work well with any text data, regardless of the class label. In the following subsections, a brief description of each category is given. However, this study focuses on supervised and unsupervised TFS models as the dominant and widely used categories.

2.5.1 Supervised Models

Supervised TFS deals with labelled document datasets [Liu and Yu, 2005, Man et al., 2009, Wang et al., 2017]. These can be collected based on specific class labels. Text data collection is usually manually labelled by domain experts. For example, a document can be positive or negative for the class label ”sport”. Class labels (or predictions/hypotheses) are of particular importance to many text mining and ML applications [Lewis et al., 2004, Li et al., 2015, Sebastiani, 2002]. They are even crucial to TFS algorithms. For example, in a given TFS technique, a class label can guide the search process for relevant features that can positively correlate with the class label [Bolón-Canedo et al., 2013, Jian et al., 2016, Li et al., 2017a].

It is also important in the definition of a relevance measure (i.e., weighting function) that differentiates features. Class labels can be utilised in generating descriptive corpus statistics [Man et al., 2009, Zhu and Lin, 2013].

As the name implies, supervised TFS models use the class information to limit the search space and evaluate features [Li et al., 2017a, Zhao et al., 2013]. Therefore, a given class label supervises the two most important tasks, namely, space searching and feature evaluation [Liu et al., 2005, Liu and Yu, 2005]. Each of these tasks can employ a so-called induction algorithm or learning algorithm to accomplish their job [Kohavi and John, 1997, Li et al., 2017b, Liu et al., 2005]. Therefore, supervised TFS techniques are further classified as filter, wrapper or hybrid as described below.

Supervised Filter Model

The vast majority of TFS algorithms follow the filter model due to its computational efficiency, scalability and generalisability [Bolón-Canedo et al., 2013, Combarro et al., 2005, Li et al., 2017a,b]. This model does not depend on any learning algorithms; instead, it selects relevant features based on the internal structure of the training dataset [Liu et al., 2005, Liu and Yu, 2005]. The supervised filter model not only depends on the characteristics of the corpus but also uses the class label to guide the search process and evaluate the extracted text features based on specific criteria [Li et al., 2017a, Liu et al., 2005]. For example, a filter model that uses the Fisher algorithm scores each feature independently based on the Fisher formula [Cai et al., 2010, Jian et al., 2016, Li et al., 2017a]. Many other techniques use different criteria for defining the relevancy of features by assigning each feature a calculated weight (or score). This score differentiates features regarding which one is more informative than the other in terms of the class label [Li et al., 2017a, Man et al., 2009]. For example, the Laplacian algorithm and spectral feature selection (SPEC) techniques belong to a family of algorithms that utilise a weights matrix analysis system known as an eigensystem to select relevant features [Cai et al., 2010, Hou et al., 2010, Wang et al., 2017, Zhao et al., 2013].

Another family of algorithms is LASSO [Tibshirani, 1996]. It attracted many researchers because it shows high performance in TFS [Li et al., 2017a, 2015]. It assigns a sparse weight to informative features, while other, non-relevant features receive a zero score. For different data structures, LASSO comes in various versions that suit them. The most influential ones are Graph Lasso, Group Lasso and Overlapping Group Lasso [Li et al., 2017a].

In the literature, most TFS algorithms belong to the filter model due to its advantages, especially its independence of any learning algorithms (classifiers). Popular algorithms like ReliefF, CFS, FCBF, t-test, Gini index, Chi-Square, IG, mRMR and many more are all good examples of TFS algorithms that follow the filter's structure [Chen and Chen, 2011, Li et al., 2017a,b, Yang and Pedersen, 1997, Zhu and Lin, 2013]. Text features produced by the filter model are more general in nature. Therefore, if the user knows the type of classifier that he or she is going to use for the text mining application, then a filter algorithm may not give accurate results as the wrapper model does.

Supervised Wrapper Model

To overcome the apparent limitations of the filter model, the wrapper method induces a classifier, and sometimes a set of classifiers, to evaluate the selected features regarding discriminating quality on the class label [Kohavi and John, 1997, Liu et al., 2005]. This makes a wrapper model a better alternative to a filter, especially when the classifier is already known beforehand. Thus, the resulting features of the wrapper model are more accurate and can lead to higher classification performance on the used classifier than those of the filter model [Bolón-Canedo et al., 2013, Liu and Yu, 2005].

First, a wrapper algorithm starts by choosing a feature subset based on some search techniques, such as the greedy search algorithm [Li et al., 2017a, Liu and Yu, 2005]. Second, the resulting features are passed to the given classifier for quality evaluation. If the features' quality is acceptable, then the wrapper stops selecting more features. Otherwise, it continues searching for another, better subset of features. This approach is computationally expensive compared with the filter model [Bolón-Canedo et al., 2013, Cai et al., 2010, Forman, 2003]. Therefore, an efficient search strategy is crucial to the success of a wrapper model.

In the relevant literature, many researchers have proposed different wrapper methods by combining different search algorithms with a variety of classifiers to achieve high-quality results [Bolón-Canedo et al., 2013, Li et al., 2017a]. For example, the SVM classifier was combined with a recursive feature elimination (RFE) search algorithm to form the so-called RFE-SVM wrapper a classification problem, and the same classifier was incorporated with the L1 norm to produce a more efficient embedded wrapper model [Li et al., 2017a, Liu and Yu, 2005].

Supervised Hybrid Model

The hybrid model has been proposed to address the obvious drawbacks of the filter and wrapper models. It combines the advantages of a filter model for being efficient in terms of choosing a subset of features and being scalable to a bigger feature space [Li et al., 2017a, Liu and Yu, 2005, Song et al., 2013]. Also, it has the evaluation accuracy of a wrapper model. This makes hybrid algorithms capable of producing a subset of features that gives a higher classification performance but that is small in feature number. Consequently, a hybrid model is considered more accurate than a filter in producing quality features, and at the same time, less computationally expensive than a wrapper [Li et al., 2017b, Liu and Yu, 2005].

As a hybrid model consists of two parts, similar to the wrapper, there are different possible combinations of classifiers and search criteria (filtering criteria) that could lead to even more efficient hybrid algorithms [Li et al., 2017b, Liu and Yu, 2005]. For example, attaching the k-nearest neighbours classifier to the combination of the correlation-based feature selection and a genetic algorithm led to a new hybrid algorithm with high accuracy [Bolón-Canedo et al., 2013, Liu et al., 2005]. Similarly, in [Li et al., 2012], combining the SVM classifier with the pattern mining algorithms has led to another new hybrid method that is capable of producing more accurate and meaningful features.

2.5.2 Unsupervised Models

In DM and ML, extracting relevant features for training classifiers requires high-quality labelled data. Providing such data for every knowledge domain is a hugely expensive task in terms of both time and cost [Algarni, 2011, Soleimani and Miller, 2016]. Further, labelling data manually is infeasible, knowing that some domain knowledge even has sub-domains, making the problem even harder. Unlabelled data, in contrast, are abundantly available for free in different information repositories, and they are growing exponentially every few months [Blei, 2012, Dhar, 2013, Khan et al., 2010]. Such data are still useful and contain invaluable knowledge that is crucial to the success of many businesses. Therefore, there is an imminent need for efficient feature selection techniques that could handle unlabelled data.

Supervised TFS methods cannot directly deal with unlabelled data due to the absence of the domain knowledge, which is represented by the class label that could guide the selection algorithm [Li et al., 2017a, Man et al., 2009]. This makes unsupervised TFS a challenging problem. To demonstrate this, imagine a collection of unlabelled text documents that have been collected from a news website, where the goal is to categorise them based on the similarity of their contents. In this case, a document can be assigned to more than one category. For example, a document with a fictitious sentence, ”An intelligent pair of shoes have been invented by Google to monitor the blood glucose level for diabetic athletes”, could be categorised into different categories like technology, health, sport, and economy. However, the absence of themes (aka topics or class labels) makes the optimal classification of the document almost impossible. TFS also cannot be done efficiently in the absence of document topics because each topic has features. In this example, the topic of technology has the feature ”Google”, while the features ”blood” and ”diabetic” belong to the topic of health and so on. Therefore, a TFS algorithm cannot efficiently calculate the feature relevancy weight (or score) in the absence of the class label.

Unsupervised TFS techniques are not as mature as supervised ones are. However, in the relevant literature, different models have been developed to tackle the problem of TFS for unlabelled documents. One commonly used approach is automatically labelling the training document set by generating topics that could be used later on to guide the TFS process and handle it as supervised ones [Blei et al., 2003, Hofmann, 2001, Wei and Croft, 2006]. One way to do this is by applying a k-means clustering technique to the training sample to generate labels [Cai et al., 2010, Hou et al., 2010, Li et al., 2017a]. Another approach is employing a spectral analysis technique to extract the underlying document clusters [Cai et al., 2010, Li et al., 2017b, Zhao et al., 2013]. An example of this technique is SPEC [Zhao and Liu, 2007], which is a unified TFS model for supervised and unsupervised data.

As can be seen so far, clustering techniques are considered a major approach for handling unlabelled data for TFS. Different k-means algorithms have been proposed to tackle this problem. For example, in, an entropy weighting k-means clustering technique has been proposed for subspace clustering [Hou et al., 2010, Liu et al., 2003]. It employs a k-means clustering algorithm to find document clusters by minimising sub-clusters in each cluster and maximising those with a negative weight. It keeps repeating these steps until it converges, and it then applies a TFS algorithm on the data. Cai et al. (2010) [Cai et al., 2010] applied spectral analysis to different features to measure the correlation between them without the need for any label information. This method, called multi-cluster feature selection, uses the top eigenvectors of a Laplacian graph to form multiple clusters. Spectral clustering can group unlabelled data without any class labels.

Clustering-based TFS techniques are not the only methods of handling unlabelled data. Other algorithms evaluate features (or terms) using a calculated weight (or score). They do not depend on any clustering techniques. For example, in text mining, TF, IDF and TF*IDF are considered the most used term-weighting functions [Salton and Buckley, 1988]. Besides, other TFS techniques cluster all features first and then select the most popular ones to be the selected features [Li et al., 2016, Song et al., 2013]. Unsupervised TFS methods follow the same categorisation system as supervised ones. They can be classified as filter, wrapper and hybrid models. These models can be developed based on the type of unsupervised technique, such as clustering [Song et al., 2013], association [], matrix factorisation [Deerwester et al., 1990] or topic modelling [Blei et al., 2003, Hofmann, 2001]. However, most models reported in the literature were developed for clustering or utilise different clustering algorithms.

Unsupervised Filter Model

The unsupervised filter model selects features based on their weights (or scores), which are assigned by the selection criteria [Cai et al., 2010, Forman, 2003, Hou et al., 2010]. Only features with higher weight are selected; those with low scores are filtered out as irrelevant or redundant. This model does not use any unsupervised learning algorithm to judge the value of the selected features [Bolón-Canedo et al., 2013, Li et al., 2017a]. Therefore, the filter model is considered fast and efficient, and it scales well with massive data. The filter model can evaluate each feature, either for its relation to the whole feature space or independently [Cai et al., 2010]. The former is called the multivariate evaluation technique, where each feature is evaluated regarding its space [Bolón-Canedo et al., 2013, Li et al., 2017a]. Thus, it is capable of finding redundant features. The latter evaluates each feature independently of the feature space, and it is known as the univariate evaluation technique [Li et al., 2017a]. This technique is much faster than the multivariate one, but it cannot handle feature redundancy [Bolón-Canedo et al., 2013].

In the literature, there are many examples of unsupervised filter algorithms. In [Hou et al., 2010], the proposed technique utilises the entropy-based distance as an evaluation criterion for the selected features, while the selection techniques in use the Laplacian score as a metric in the evaluation and selection tasks. SPEC [Zhao and Liu, 2007] is considered a univariate model, and it has been extended to work as a multivariate technique. Finally, the filtering algorithm in [Hou et al., 2010] implements a feature-dependency metric for measuring the feature relevancy score.

Unsupervised Wrapper Model

This type of wrapper model is similar in its internal structure to the supervised one, except it utilises an unsupervised classifier (e.g. clustering algorithm) [Hou et al., 2010, Li et al., 2017a, Liu and Yu, 2005]. First, the model starts by selecting a subset of features, and then it passes them to the unsupervised classifier. Second, the classifier evaluates those features based on their discriminating quality. If they can form better clusters, for example, then the algorithm stops. Otherwise, the model repeats the first and second steps until it produces the best possible clusters [Dy and Brodley, 2004, Li et al., 2017a]. This makes the wrapper model highly computationally expensive compared with the filter model, as it tries to evaluate all the available subsets of features. This can be a prohibitive task, especially with a high-dimensional dataset [Bolón-Canedo et al., 2013]. One possible solution to this problem is reducing the total search space in the sample features by implementing a more efficient search algorithm, such as the heuristic method [Peng et al., 2005]. In addition, there is another problem with the wrapper model: It can be biased to the chosen unsupervised classifier [Li et al., 2017b]. However, the wrapper model gives better features compared with the filter model because it selects only features that form high-quality clusters.

In the literature, many different TFS techniques have been built on the wrapper model. However, they differ in their search strategy and the unsupervised learning algorithm. For example, the classical k-means clustering algorithm has been used as a classifier, and it can be accompanied by any search technique as a feature selector [Li et al., 2017a]. A second wrapper example utilises Gaussian methods as unsupervised classifiers and maximum-likelihood criteria for selecting the subset of features [Cai et al., 2010]. A final example was reported in [Dy and Brodley, 2004], where the authors used the expectation maximization (EM) clustering technique to group the selected features and then evaluated the resulting clusters based on specific criteria. This technique is called feature subset selection wrapped around EM clustering.

Unsupervised Hybrid Model

A hybrid model was developed to tackle the apparent limitations of the unsupervised filter and wrapper algorithms. This model tries to combine an efficient filtering method with a suitable unsupervised learning algorithm (or unsupervised classifier) to produce a better subset of features [Li et al., 2017a, Liu et al., 2005, Liu and Yu, 2005]. Typically, a hybrid model starts by selecting different subsets of features using its filtering criteria. Then, it evaluates the quality of each subset individually by passing them sequentially to the unsupervised classifier (e.g. classical k-means clustering algorithm) [Cai et al., 2010, Liu and Yu, 2005]. Finally, the model selects only one subset that has the highest quality (or produces best clusters) [Cai et al., 2010, Li et al., 2017a]. Clearly, the hybrid model is more efficient than the wrapper model in terms of speed and quality results, but it is slower than the filter model.

2.6 Feature Selection Applications

Due to their benefits discussed above, many TFS models and frameworks have been extensively used in different applications of text-based information analysis. As shown in Figure 2.10, there are many applications in which TFS can be used. Popular applications are text classification, text clustering, text summarisation, information retrieval/filtering, natural language processing, text visualisation, social media analysis and others. This study briefly discusses the use of TFS with the applications depicted in the figure and how essential was the role of TFS in helping these applications to achieve their goals.

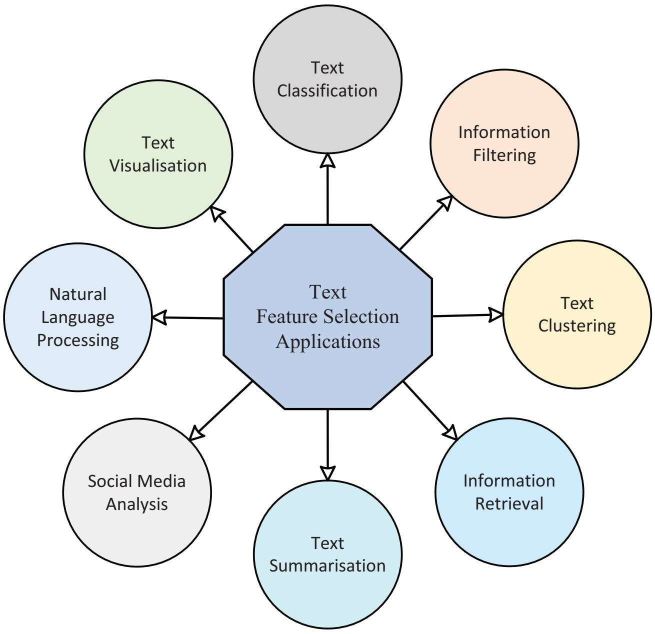


Figure 2.10: TFS applications.

- **Text Classification**: Supervised text classification (aka text categorisation) is the task of automatically assigning text documents to a predefined category (i.e., class or label) [Forman, 2003, Khan et al., 2010]. Supervised and unsupervised TFS has been extensively used with various text classifiers, such as SVM, k nearest neighbours, Rocchio, Naive Bayes, decision tree, neural networks and many other variations of these classifiers [Aphinyanaphongs et al., 2014, Khan et al., 2010, Li et al., 2017c, Yang and Pedersen, 1997]. Commonly, and before the training phase of a text classifier, a TFS method is used to select a small set of informative features from the labelled training collections to reduce the total dimensionality of the feature space in the collection. The selected features can then be used to represent the training documents that the text classifier will be trained on [Aphinyanaphongs et al., 2014, Li et al., 2017c]. The non-selected features can be removed with minimal effects on the overall accuracy of the classification algorithms. Different experimental studies in [Aphinyanaphongs et al., 2014, Chen et al., 2016, Escalante et al., 2015, Li et al., 2017c, Liu et al., 2009] clearly demonstrate that TFS is an essential step for the effectiveness and efficiency of text classifiers.

- **Text Clustering**: Despite being an unsupervised learning task that groups similar pieces of text together [Anastasiu et al., 2013, Jain, 2010], text clustering still requires TFS to identify and remove noisy features as well as reduce the total dimensionality of the feature space [Aggarwal and Zhai, 2012, Liu et al., 2003]. Text clustering is used extensively to find interesting patterns from unlabelled collections of documents using some similarity functions, and, then, organise these documents to improve tasks like retrieval, filtering, summarisation and others [Aggarwal and Zhai, 2012, Alharbi et al., 2017b, Huang, 2008, Jain, 2010, Liu and Croft, 2004]. However, the absence of the class label from the used collection has made the selection of informative features more challenging. In such a case, supervised TFS techniques might not be applicable, and only unsupervised TFS can be used for text clustering. TFS methods, such as TF, TFIDF, LSA, PLSA, LDA and others have been widely used with text clustering [Anastasiu et al., 2013, Liu et al., 2003, Shehata et al., 2010, Wang et al., 2015]. The experimental studies in [Beil et al., 2002, Lee et al., 2015b, Liu et al., 2003, Shehata et al., 2010] show that TFS not only improves the efficiency of clustering algorithms but also leads to higher clustering performance.

- **Text Summarisation**: TFS models play an essential role in multi-document summarisation despite the underlying algorithms [Qiang et al., 2016]. Text summarisation intends to automatically produce a concise and coherent summary that must retain the key information in the original text [Qiang et al., 2016, Wu et al., 2019]. TFS models are extensively employed with both extractive and abstractive methods of text summarisation [Qiang et al., 2016, Wu et al., 2019, 2016]. Unsupervised term-based TFS models, such as TF, TFIDF and other frequency-based schemes, are efficiently used to select informative terms, and, thus, indicate candidate sentences to the summarisation algorithms [Qiang et al., 2016, Wu et al., 2019]. Supervised term-based models are also used in text summarisation, especially when training samples are available. However, the lack of semantic information in low-level terms has limited their use and moved the focus towards semantically-rich text features. Therefore, closed patterns and a combination of patterns and latent topics have produced better summarises despite their time-complexity [Qiang et al., 2016, Wu et al., 2019]. A more holistic approach is also taken through the combination of low-level terms and high-level patterns and topics, which gives better summarisation performance [Wu et al., 2019].

- **Information Retrieval**: For decades, different types of TFS models are employed by many IR models. Generally, IR concerns about locating relevant information from a collection of documents given a query that represents user information need [Belkin and Croft, 1992, Croft, 2000, Gao, 2015]. Term-based TFS models received much attention in the IR community due to their efficiency and mathematical soundness. Popular term-based models used in IR are TFIDF [Salton and Buckley, 1988], Rocchio [Rocchio, 1971], BM25 [Robertson and Zaragoza, 2009], Prob [Jones et al., 2000a,b], SVM [Joachims, 2002] and many more. However, the sensitivity of these models towards semantic-related problems has impacted their performance in retrieval tasks [Gao, 2015, Li et al., 2015, Metzler, 2007]. Thus, TFS techniques that adopt high-level features are widely used in retrieval models instead of the term-based to tackle the problems of information mismatch and overload. The n-grams statistical language model [Bendersky and Kurland, 2010, Lavrenko and Croft, 2001] is of a particular interest and shows better retrieval results compared to the traditional term-based techniques. As the user's query and the collection might exhibit multiple topics, different statistical, topic-based TFS models are also used in IR. Models like LSA [Deerwester et al., 1990], PLSA [Hofmann, 2001] and LDA [Blei et al., 2003] and its variations are intensively used to reduce the impact of synonymy and polysemy problems. However, topic-based models are limited in terms of their semantic capabilities. Thus, concept-based TFS models that use external knowledge bases, such as Wikipedia, Wordnet, DBpedia, etc. are also used in retrieval models to add explicit semantics to the document representation, and, thus, solve the synonymy and polysemy problems [Bendersky et al., 2011, Egozi et al., 2008].

Nevertheless, the expensive time-complexity of these models and the effectiveness of these external resources in a real retrieval system are limiting their use. Overall, and based on the above studies, it can be concluded that TFS continues to contribute to the success of many IR applications.

- **Information Filtering**: As TFS plays a crucial role in the IR field, it is expected to continue the same role in IF because both fields seek to find relevant information that suit user information needs [Belkin and Croft, 1992, Gao, 2015, Robertson and Soboroff, 2002]. However, unlike IR, IF dynamically removes irrelevant stream of documents based on maintained user information needs (aka user profiles or long-term user interests) [Li et al., 2012, Robertson and Soboroff, 2002, Soboroff and Robertson, 2003]. Different TFS models and frameworks are used to select relevant features from a collection of documents that discusses user information preferences. Similar to the IR context, the conventional term-based TFS models are widely used in IF, particularly the supervised models, such as SVM [Li et al., 2010], MI [Manning et al., 2008b], Chi-square [Chen and Chen, 2011], BM25 [Robertson and Zaragoza, 2009], Rocchio [Rocchio, 1971], LASSO [Tibshirani, 1996] and many others. As term-based TFS models do not consider the order of terms in the documents, phrase-based methods are employed, especially the n-grams-based models [Albathan et al., 2013, Fürnkranz, 1998], because phrases carry more semantic information than low-level terms. Data mining approaches are also used in IF to reduce the effects of synonymy and polysemy problems, most notably the pattern-based TFS techniques like PTM [Wu et al., 2004], PDS [Zhong et al., 2012], MP [Yan et al., 2005], RFD [Li et al., 2015], PCM [Albathan et al., 2012] and SCSP [Albathan et al., 2014]. However, all these TFS models do not assume that user information interests can span many topics and themes [Blei et al., 2003, Gao et al., 2015]. Thus, topic-based TFS models like PLSA [Hofmann, 2001], LDA [Blei et al., 2003] and their extensions are adopted by IF systems to handle the multiple topics assumption in the documents that describe user information preferences. Nevertheless, and on the basis that no single feature can hold all relevant information, different hybrid-based TFS models and techniques are used for IF. Popular examples are the pattern-based topic models (e.g., PBTM-FP [Gao et al., 2013], PBTM-FCP [Gao et al., 2013], SPBTM [Gao et al., 2014b] and MPBTM [Gao et al., 2015]), the ontological concept-pattern [Bashar and Li, 2018, Bashar et al., 2017] and concept-topic models [Bashar and Li, 2017, Bashar et al., 2016] approaches. The experimental studies in all these previously mentioned models clearly show the important role of TFS in IF and how some features are more informative than the others.

2.7 Chapter Summary

This chapter provided an in-depth discussion of TFS techniques in extant literature. First, the discussion encompassed knowledge discovery in databases in which feature selection plays an important role. Next, the discussion outlined text mining and the techniques of text representation and pre-processing. The discussion then focussed on conventional TFS approaches along with highlighting their challenges and issues. However, it was determined that the presence of uncertainties is still a challenging problem, causing relevant features to be missed, overestimated or underestimated.

The next chapter introduces our innovative and effective SIF model. The model adopts a hybrid fusion strategy of different lexical and statistical features that are discovered from a set of relevant documents that discuss user information needs. Our SIF model extends multiple ERSs to accurately estimate the relevance of topical terms that occur across the documents. The chapter also presents essential definitions in relation to random-sets, topic modelling, global statistics and the specifically defined text feature fusion.

Chapter 3

Fusion Model for Relevant Feature Selection

3.1 Introduction

As noted in Chapter 1, TFS has been extensively researched by many communities due to its importance to a broad spectrum of applications [Blei et al., 2010a, Forman, 2003, Li et al., 2017a, Man et al., 2009, Yang and Pedersen, 1997, Zhao and Liu, 2007]. In relevance discovery, selecting features from the contents of a long document set that describes user information needs is difficult, due to the uncertainties in these documents [Li and Zhong, 2003, Li et al., 2005, Zhong et al., 2012, Zhou et al., 2011]. The selection problem becomes more challenging in the absence of a user's query that could guide the search for relevant features. However, in IR, fusion-based techniques have shown remarkable results in identifying relevant documents compared to traditional models in the field [Lillis et al., 2006, 2010]. These techniques show that combining different representations of documents and queries, search system outputs, and ranking and scoring algorithms as evidence of relevance can reduce uncertainty and yield better results [Croft, 2000, Kozorovitsky and Kurland, 2011a]. However, adopting a similar approach for TFS for relevance discovery is difficult, because which text features to fuse, how to fuse them effectively and, ultimately, how to use the fusion feature to manage uncertainties in the relevant document set remains unknown.

Random sets are effective mathematical tools for handling uncertainty and vagueness in an information system [Goutsias et al., 1997, Molchanov, 2005, Nguyen, 2008] and can be extended to an ERS [Albathan et al., 2014, Li, 2003]. An ERS can be used to describe interesting relationships that are inherited between conditional and decisional entities [Li, 2003, Li and Yao, 2002b], which makes it the best fit for modelling and managing the fusion of text features based on their importance to different entities in a document collection. User information needs can implicitly cover several topics, as illustrated in Figure 1.2, and a long document (Figure 1.1) that is relevant to the user information needs can also discuss multiple subtopics or themes in its segments (e.g., paragraphs or sentences). Traditional TFS methods do not assume that long documents exhibit multiple topics—they also have no mechanism to discover them or determine their relevance to the user information needs [Gao et al., 2014b, 2015]. Topic-based methods, such as PLSA [Hofmann, 2001] and LDA [Blei et al., 2003], have been explicitly built to treat documents as a mix of latent topics. Unlike PLSA, LDA is more popular with many applications Blei et al. [2010a], Blei [2012]. In the form of a language-modelling approach, LDA can also adopt different document representations and accurately estimate features' probabilities [Blei et al., 2003, Croft, 2000, Gao et al., 2015], which makes the topical features of the LDA better candidates for an effective fusion-based TFS.

Analogous to IR fusion-based techniques that reward highly ranked documents in retrieved lists [Anava et al., 2016, Lillis et al., 2006], TFS models for relevance discovery must also reward highly relevant features. The most critical component in any TFS model is thus the weighting function [Li et al., 2015, 2010, Wu et al., 2006]. It assigns a numerical weight to each feature, specifying how informative the feature is to the user information needs [Albathan et al., 2013, 2014, Li et al., 2015]. However, LDA estimates a term[1] weight that is locally based on two components: the topic–document distribution and the term–topic assignment [Blei et al., 2003, Gao et al., 2015]. Therefore, in a set of similar documents, a specific term might receive a different weight in each separate document, even though the term is semantically identical across all the documents. Such an approach does not accurately reflect the semantic meaning and relevance of the term to the user information needs. The performance of LDA in TFS for relevance discovery is influenced negatively, as it is uncertain and difficult to know which weight is more representative; it should thus be assigned to the intended relevant term. Several experiments in different studies [Alharbi et al., 2017b,c, Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2013, 2014b, 2015] confirm that the term probability (i.e., weight) function of LDA make it ineffective in discovering relevant topical terms.

Identifying relevant terms from a collection of documents that describe user information needs can be achieved by combining evidence about these terms in different representations [Lee et al., 2015a, Zhang and Balog, 2017]. The global statistics of terms, such as document frequency (df), term frequency (tf), paragraph frequency (pf) or inverse document frequency

[1] In this chapter, terms, words, keywords or unigrams are used interchangeably.

(idf) are important evidence that reveal the discriminatory power of terms [Man et al., 2009, Maxwell and Croft, 2013]. However, in IR, selecting terms that are based on global statistics did not show better retrieval performance [Bendersky et al., 2011, Macdonald and Ounis, 2010], because global statistics cannot describe the local importance of terms [Maxwell and Croft, 2013]. Inversely, LDA can estimate the local importance of terms at the document level based on the two components that were mentioned previously. However, LDA estimates the term–topics probabilities globally, which does not correctly reflect the importance of the term at the document level because terms usually appear unevenly across the relevant document set. Therefore, fusing the different weights of the LDA topical features in a global context is challenging and remains uncertain due to the complex relationships between terms and different entities that represent the document collection. For example, a term might appear in multiple documents, paragraphs and LDA topics. Similarly, each topic might be discussed, entirely or partially, in many documents or paragraphs that contain the same term.

This chapter presents SIF,[2] a novel and effective fusion-based TFS model for relevance discovery. SIF is proposed to solve the previous questions and overcome the common limitations of the existing TFS models of relevance discovery (i.e., a more accurate estimation of the relevant features). The proposed model is derived from random set theory to handle uncertainties, manage features fusion and model the complex relationships between essential entities in a set of relevant documents. Figure 3.1 shows the SIF's structure, with the feature fusion module at the core of the model, and the main entities—namely, the paragraph, term and topic sets—that are adopted from the relevant document collection D^+. Further, the flow of different lexical (i.e., terms) and statistical features are also depicted. The remainder of this chapter provides basic definitions about topic modelling, LDA, global statistics, text feature fusion and random sets in Section 3.2; more details about the SIF model are presented in Section 3.3 and a summary in Section 3.4. An extensive evaluation of the proposed SIF model is presented in Chapter 6.

3.2 Basic Definitions

Given a document collection D, the relevant long documents set $D^+ \subseteq D$ discusses user information needs that might contain multiple topics of interest. Notably, in the current study, a topic of interest is different to a latent topic that can be discovered by a topic modelling

[2] This model was published in [Alharbi et al., 2017c] and the acronym **SIF** stands for **S**election of **I**nformative **F**eatures.

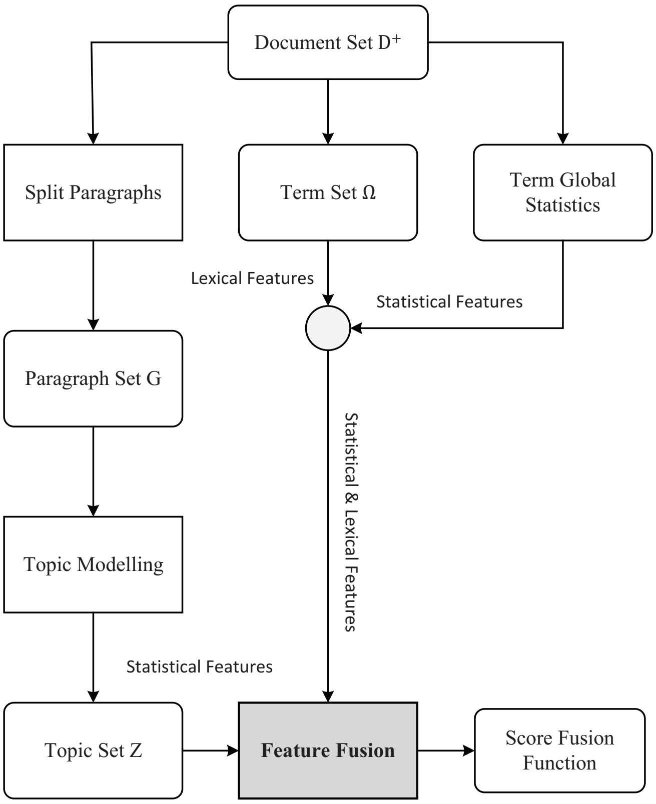


Figure 3.1: The SIF model structure.

algorithm (e.g., LDA). The SIF model uses D^+ for training, whereby each document $d_x \in D^+$ has a set of paragraphs S and each paragraph has a set of terms. The set G in this thesis is the set of all paragraphs in D^+, such that $S \subseteq G$. The set of terms Ω denotes the vocabulary list in D^+. A term t is a keyword or unigram in which the function $terms(g)$ returns the set of terms that appear in paragraph g.

In the proposed SIF model, the paragraphs of relevant documents are split and each paragraph is treated as an independent passage (i.e., a document) that consists of a bag of terms (i.e., words), as illustrated in Table 3.1 in which a term t_i, for example, may appear more than once in a paragraph g_y. Before delving into the details of SIF, the next section provides some essential definitions of topic modelling and the LDA, followed by global statistics and feature fusion in this thesis, including its strategies.

3.2.1 Topic Modelling

Topic modelling algorithms, such as PLSA [Hofmann, 2001] and LDA [Blei et al., 2003], are proven to be effective in reducing the total dimensionality of text documents to a set of manageable topics [Gao et al., 2015]. LDA is more effective than PLSA in many applications

Table 3.1: A sample of document collection with three hierarchical entities: document, paragraph and term

Document	Paragraph	Term
d_1	g_1	$\{t_1, t_2, t_4, t_3, t_7, t_2\}$
	g_2	$\{t_3, t_1, t_5, t_7, t_1\}$
	g_3	$\{t_5, t_6, t_4, t_2, t_1, t_2, t_7\}$
	g_4	$\{t_4, t_2, t_3, t_7\}$
d_2	g_5	$\{t_1, t_3, t_4, t_3, t_8, t_2\}$
	g_6	$\{t_3, t_2, t_5, t_7\}$
d_3	g_7	$\{t_4, t_6, t_8, t_2, t_1, t_5, t_7\}$
	g_8	$\{t_1, t_2, t_3, t_7\}$
	g_9	$\{t_3, t_8, t_5, t_7, t_3\}$

[Blei, 2012, Gao et al., 2014b, 2015] and can statistically identify hidden topics from a text collection to improve different tasks in IR [Wang et al., 2007, Wei and Croft, 2006], IF [Gao et al., 2015], document summarisation [Wu et al., 2016], visualisation [Chaney and Blei, 2012], personalised ontology learning [Bashar et al., 2016] and many other TM and ML applications. LDA represents documents by a set of topics in which each topic is a set of semantically related terms [Blei et al., 2003, Gao et al., 2015]. It can thus group related words in a document collection, which can reduce the negative influence of common problems like polysemy, synonymy and information overload [Aggarwal and Zhai, 2012, Gao et al., 2014b]. However, in practice, LDA treats topics as multinomial distributions over words and represents documents as a probabilistic mix over a predefined number of latent topics. LDA is discussed further in the next section.

3.2.1.1 Latent Dirichlet Allocation

Given the set of relevant documents D^+, the proposed SIF model uses LDA to reduce the total dimensionality of D^+ paragraphs to a set of manageable topics Z, in which V denotes the number of topics in Z. Therefore the input to LDA in our study is the set of all paragraphs G, as illustrated in Table 3.2. Splitting the paragraphs of the long documents before the topic discovery step implicitly exploits the relationships between terms that commonly appear in similar contexts [Krikon and Kurland, 2011, Xi et al., 2001]. Moreover, LDA assumes that each paragraph has multiple latent topics and that each topic $z_j \in Z$ is defined as a multinomial probability distribution over all terms in Ω, as shown in Table 3.3, that are represented as

$p(t_i|z_j)$, in which $t_i \in \Omega$ and $1 \leq j \leq V$ such that

$$\sum_i^{|\Omega|} p(t_i|z_j) = 1$$

Table 3.2: A set of paragraphs of the documents in Table 3.1 and their terms, which both represent the input to be given to LDA

Paragraph	**Terms**
g_1	$\{t_1, t_2, t_4, t_3, t_7, t_2\}$
g_2	$\{t_3, t_1, t_5, t_7, t_1\}$
g_3	$\{t_5, t_6, t_4, t_2, t_1, t_2, t_7\}$
g_4	$\{t_4, t_2, t_3, t_7\}$
g_5	$\{t_1, t_3, t_4, t_3, t_8, t_2\}$
g_6	$\{t_3, t_2, t_5, t_7\}$
g_7	$\{t_4, t_6, t_8, t_2, t_1, t_5, t_7\}$
g_8	$\{t_1, t_2, t_3, t_7\}$
g_9	$\{t_3, t_8, t_5, t_7, t_3\}$

Table 3.3: A sample of LDA topics generated from collection 101 of the RCV1 dataset, which shows how LDA represents a latent topic (i.e., a probability distribution over terms)

Topic 1	**Topic 3**	**Topic 5**	**Topic 10**
year 0.072	Piech 0.148	Volkswagen 0.086	VW 0.264
federal 0.072	economic 0.055	passed 0.06	house 0.059
AG 0.04	manager 0.047	laws 0.043	sale 0.045
seat 0.032	interview 0.032	million 0.035	theft 0.045
economic 0.032	use 0.032	Europe 0.035	GM 0.037
board 0.032	congress 0.032	USA 0.035	test 0.037
believed 0.024	computer 0.032	prosecutor 0.035	fight 0.023
work 0.024	return 0.024	version 0.026	appeared 0.015
suspect 0.024	organisation 0.024	Lopez 0.026	full 0.015
advances 0.024	agency 0.024	planted 0.018	lose 0.015

LDA also represents an individual paragraph g as a probabilistic mixture of topics as $p(z_j|g)$, as illustrated in Table 3.4. Consequently, and based on the number of latent topics, the probability (i.e., local weight) of term t_i in paragraph g can be calculated as

$$p(t_i|g) = \sum_{j=1}^{V} p(t_i|z_j) \times p(z_j|g)$$

All hidden variables, $p(t_i|z_j)$ and $p(z_j|g)$, are statistically estimated by the Gibbs sampling

Table 3.4: Example of how LDA represents a paragraph (i.e., a probability distribution over latent topics)

Paragraph	$Z_1(\vartheta_{y,1})$	$Z_2(\vartheta_{y,2})$	$Z_3(\vartheta_{y,3})$	$Z_4(\vartheta_{y,4})$
g_1	0.2	0.3	0.4	0.1
g_3	0.4	0.3	0.1	0.2
g_5	0.1	0.2	0.5	0.2
g_6	0.3	0.2	0.2	0.3
g_8	0.1	0.6	0.1	0.2

algorithm [Steyvers and Griffiths, 2007].

From a different perspective, LDA generates two distinct outputs that can be observed from two levels. At the document level (or, in our case, the paragraph level), LDA represents each paragraph g_y by the proportions of topics distribution $\theta_{g_y} = (\vartheta_{y,1}, \vartheta_{y,2}, \vartheta_{y,3}, \ldots, \vartheta_{y,v})$. At the collection level, which in our model is the set of relevant documents D^+, LDA represents D^+ by a set of topics Z, in which each topic is a probability distribution over all terms in D^+, ϕ_j for topic z_j and $\Phi = \{\phi_1, \phi_2, \phi_3, \ldots, \phi_v\}$ for all topics. Different studies [Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2014b, 2015] commonly only use the top 10 terms from each topic, based on their probability distribution that is estimated by $p(t|z)$. However, the proposed model considers all terms in all topics to avoid any possibility of relevant feature loss during the training phase of the SIF model. A third output that the LDA can produce is the term–topic assignment, in which a set of terms is assigned to a specific topic but not to other related topics.

3.2.2 Global Statistics

Global statistics are frequency-based evidence that are used to indicate the informativeness (i.e., importance) of text features (e.g., terms and phrases) to the entire collection of documents [Macdonald and Ounis, 2010, Maxwell and Croft, 2013]. In IR, and by using the global statistics of terms, documents are usually scored and then ranked according to the presence and/or frequency count of query terms in the document [Macdonald and Ounis, 2010, Maxwell and Croft, 2013, Sebastiani, 2002]. However, global statistics do not provide semantic information about terms and can be biased towards the most frequent terms in the collection, which are often general and less discriminating [Bendersky et al., 2011, Maxwell and Croft, 2013]. They thus do not show significant improvement in IR, especially in weighting models of proximity [Huston and Croft, 2014, Macdonald and Ounis, 2010, Maxwell and Croft, 2013]. For relevant feature discovery, global statistics show how terms are explicitly related to relevance at the collection level, but they ignore the local (i.e., at document level) details of relevance [Macdonald and Ounis, 2010, Maxwell and Croft, 2013], as illustrated in Table 3.5 for the case of document frequency df. Additionally, global statistics cannot deal with latent information (e.g., latent topics) alone, and they ignore explicit semantic relationships between terms [Bendersky et al., 2011]. In the current study, we divide global statistics into two groups: raw statistics and handcrafted statistics.

Table 3.5: Document frequency of terms (df) for a set of relevant long documents in which the local semantic and statistical details of each term in any document are ignored, except for the presence and absence of terms in these documents

Document	t_1	t_2	t_3	t_4	t_5	t_6	t_7
d_1	1	1	1	0	0	1	1
d_2	0	1	1	1	0	1	1
d_3	1	0	0	0	1	0	1
d_4	0	1	1	0	0	0	1
d_5	0	1	0	0	0	1	1
d_6	1	1	0	1	0	1	1
df	**3**	**5**	**3**	**2**	**1**	**4**	**6**

3.2.2.1 Raw Statistics

As the name indicates, these statistics are non-estimated and primarily based on the count of the terms in the document collection. Popular examples are the df, tf and other segment-based frequencies, like pf and sentence frequency (sf). Each of these global statistics is characterised by identifying the discriminating power of a term [Macdonald and Ounis, 2010]. Therefore, document frequency and term frequency form the basis for other handcrafted global statistics—like idf [Salton et al., 1975] and $tfidf$ [Salton and Buckley, 1988], which are discussed in the next section. However, they can either underestimate or overestimate the importance of terms to the user information needs [Macdonald and Ounis, 2010, Maxwell and Croft, 2013]. Given the collection of relevant documents D^+, we define these raw statistics in this research as follows:

Definition 1 (Document Frequency) *The df of a term t is the number of relevant documents in D^+ that contain the term t. $df(t)$ can be calculated as follows:*

$$df(t) = \sum_{i=1}^{|D^+|} f_{d_i}(t)$$

where $|D^+|$ is the total number of relevant documents in D^+ and $f_{d_i}(t)$ can be defined as follows:

$$f_{d_i}(t) = \begin{cases} 1, & \textit{if } t \in d_i \\ 0, & \textit{otherwise} \end{cases}$$

Definition 2 (Paragraph Frequency) *The pf of a term t is the number of paragraphs in D^+ that contain the term t. Thus, given G, the set of all paragraphs in D^+, $pf(t)$ can be calculated in a similar manner as df using the following formula:*

$$pf(t) = \sum_{y=1}^{|G|} f_{g_y}(t)$$

where $|G|$ is the total number of paragraphs in G and $f_{g_y}(t)$ is defined like $f_{d_i}(t)$ as follows:

$$f_{g_y}(t) = \begin{cases} 1, & \textit{if } t \in g_y \\ 0, & \textit{otherwise} \end{cases}$$

Definition 3 (Term Frequency) *The tf of a term t is the number of times t occurs over D^+. Unlike the definitions of df and pf, which are only concerned about the binary occurrence (i.e., occur or does not occur) of the term t in a relevant document or paragraph, $tf(t)$ also considers redundant occurrences of t. $tf(t)$ can thus be calculated as follows:*

$$tf(t) = \sum_{i=1}^{|D^+|} fr(t, d_i)$$

where $fr(t, d_i)$ counts the total occurrences of term t in document d_i

3.2.2.2 Handcrafted Statistics

Handcrafted statistics are developed and estimated to suit different needs based on counting term occurrences within a collection of documents. Widely used examples are the idf and term frequency–inverse document frequency ($tfidf$), which are essential components in many term-weighting models, particularly in IR, such as BM25 [Robertson and Zaragoza, 2009], language models and smoothing formulas [Zhai and Lafferty, 2004] for estimating the relevance of documents given a user query. In this research, we define the cluster frequency of terms as a handcrafted statistic, in addition to idf and $tfidf$ as follows:

Definition 4 (Cluster Frequency) *Given C, as a set of document clusters in D^+, the cluster frequency (cf) of a term t is the number of document clusters that contain the term t. The term t occurs in a cluster $c_j \in C$ if at least one relevant document in that cluster contains t. If we assume that D^+ has L clusters, then $cf(t)$ can thus be calculated as follows:*

$$cf(t) = \sum_{j=1}^{L} f_{c_j}(t)$$

where $f_{c_j}(t)$ can be defined as follows:

$$f_{c_j}(t) = \begin{cases} 1, & \textit{if } t \in c_j \\ 0, & \textit{otherwise} \end{cases}$$

Definition 5 (Inverse Document Frequency) *The idf is developed to quantify the informativeness of a term t in D^+, assuming that the important terms appear less frequently in fewer documents than the less-important terms. Using the previous definition of $df(t)$, the formula of idf for term t can be expressed as follows:*

$$idf(t) = log\left(\frac{|D^+|}{df(t)}\right)$$

Definition 6 (Term Frequency–Inverse Document Frequency) *The $tfidf$ combines a local $tf(t)$ in document d as a representative statistic of the document's contents, with the global $idf(t)$ as a discriminating statistic. While there are many variants of $tfidf$ [Sebastiani, 2002], the current study estimates it to be in accordance with Salton and Buckley [1988], as follows:*

$$tfidf(t) = fr(t,d) \times idf(t)$$

Moreover, using global statistics alone can impose inflexibility, as each global statistic has its own limited focus and does not consider other factors that influence term weight. For example, in IR, idf was introduced to control the effect of frequent terms in the collection based on the assumption that less frequent terms are more specific [Cummins and O'riordan, 2005, Macdonald and Ounis, 2010]. However, idf cannot look beyond the importance of document frequency to terms [Bendersky et al., 2011]. These limitations make utilising global statistics alone for term weighting ineffective due to their single, low-level focus. Therefore, there is a need to fuse informative global statistics with other high-level features to (1) increase the flexibility of such statistics, (2) enhance the representativeness of features, and (3) accurately estimate the importance of other text features.

3.2.3 Text Feature Fusion

Text feature fusion can be defined as the process of integrating different lexical, syntactic, semantic and statistical features into a useful, consistent and accurate representation of text documents [Balazs and Velásquez, 2016, Egozi et al., 2008, Scott and Matwin, 1999, Wu et al., 2006, Xue and Zhou, 2009]. This fused representation is then used to enhance the performance of the related TM and ML tasks (e.g., retrieval [Anava et al., 2016, Pickens and Golovchinsky, 2008], filtering [Alharbi et al., 2018b, Gao et al., 2015], classification [Bharath Bhushan and Danti, 2017, Xu et al., 2017] and clustering [Wang et al., 2019, Yu et al., 2011]), because no specific feature can hold the available pieces of evidence information singlehandedly. However, as previously mentioned, it is challenging to know which type of text features to fuse, how to fuse it and, ultimately, how to weight and select the most relevant text features from the contents of relevant documents that describe user information needs, knowing that plain text is monomodal and suffers from inherited problems like synonymy, polysemy, noise, feature sparsity and many uncertainties [Gao et al., 2015, Jian et al., 2016, Li et al., 2015, Zhong et al., 2012].

A fusion-based TFS technique might first exploit inter-feature semantic relationships—such as dependency [Chen et al., 2017, Xu et al., 2017], co-occurrence [Balazs and Velásquez, 2016, Kludas, 2011], correlation [Kim et al., 2010], causation [Xiao et al., 2016] and mutual information [Peng et al., 2005]—as evidence to locate interesting features [Kludas, 2011, Wu and Mcclean, 2006]. It could then combine their normalised scores (i.e., weights) or relevance rankings before estimating the features' final scores [Lillis et al., 2006, 2008, Nuray and Can, 2006]. The feature-scoring function is thus critical in the fusion-based TFS algorithm and should estimate an informative score to each fused feature [Bendersky et al., 2011, Li et al., 2015]. Despite the single modality of text, its feature fusion algorithms can still be divided into the typical three groups of fusion strategies—early, late and hybrid—as in multimodal ML [Alqhtani et al., 2018, Baltrušaitis et al., 2019]. However, in our research, the presence and absence of semantic information in the used text features will govern the distinction between those strategies.

3.2.3.1 Early Fusion

In this thesis, a TFS model can adopt the early fusion strategy if it only integrates individual terms by combining their scores (e.g., frequencies) heuristically *before* and considering any semantic relationships between these terms and relevant and/or irrelevant documents. In relevant feature discovery, the traditional term-based techniques, such as idf [Salton et al., 1975], tfidf Salton and Buckley [1988], BM25 [Robertson and Zaragoza, 2009], Prob [Jones et al., 2000a,b], SVM [Joachims, 2002], χ^2 [Chen and Chen, 2011], MI [Manning et al., 2008b], LASSO [Tibshirani, 1996] and Rocchio [Rocchio, 1971], are popular examples of early fusion-based TFS models. Therefore, we generally define the early fusion of text features as follows:

Definition 7 (Early Fusion) *A fusion strategy that integrates low-level terms before considering any form of semantic relationships between them and relevant and/or irrelevant documents.*

Moreover, in multimodal fusion, early fusion strategy is commonly used and known as the feature-level fusion [Atrey et al., 2010, Datta et al., 2017, 2016] or pre-classification fusion [Jeng and Chen, 2016]. It is claimed that the early fusion of features is best in terms of performance improvement, as the original raw source of information is considered the richest [Balazs and Velásquez, 2016, Kludas, 2011, Zhang and Balog, 2017]. However, in most cases, low-level terms in text data suffer from inherited noise and cannot handle semantic-related problems (e.g., synonymy and polysemy) because they ignore the order of the terms in documents [Li et al., 2010, 2012].

3.2.3.2 Late Fusion

High-level text features, such as phrases (n-grams), patterns, topics, concepts or a combination of these, are more semantically rich than low-level individual keywords (i.e., terms) [Bashar et al., 2016, Gao et al., 2015, Li et al., 2015, Tao et al., 2011, Zhong et al., 2012]. Therefore, documents that share the same high-level features are more likely to be semantically related [Gao et al., 2015, Zhong et al., 2012]. In this research, and from a data fusion perspective, a TFS model can apply the late fusion strategy if the features' scores are combined *after* the extraction of some high-level features from relevant and/or irrelevant documents. Popular examples of late fusion–based TFS models are the phrase-based n-grams models [Fürnkranz, 1998, Lavrenko and Croft, 2001]; the pattern-based PTM [Wu et al., 2004], PCM [Albathan et al., 2012] and SCSP [Albathan et al., 2014] models; the topic-based PLSA [Hofmann, 2001] and LDA [Blei et al., 2003] models; the concept-based CBM [Shehata et al., 2007] model; and other hybrid, high-level features-based models, such as the pattern– topic based PBTM-FP [Gao et al., 2013], PBTM-FCP [Gao et al., 2013], SPBTM [Gao et al., 2014b] and MPBTM [Gao et al., 2015] models; the topic–phrase TNG [Wang et al., 2007] model; and the pattern–concept PIM [Bashar and Li, 2018] and topic–concept Lda Concept [Chemudugunta et al., 2008] models. Therefore, we define the late fusion of text features as follows:

Definition 8 (Late Fusion) *A fusion strategy that integrates high-level features after considering any form of semantic relationships between them and relevant and/or irrelevant documents.*

In multimodality, the late fusion strategy is also known as decision-level fusion [Atrey et al., 2010, Datta et al., 2017, Kludas, 2011] or post-classification fusion [Alqhtani et al., 2018]. In this approach, the fusion occurs at the concept level [Alqhtani et al., 2018] because some evidence-based decisions have already ocurred by merely combining the feature scores from each learning model [Alqhtani et al., 2018, Baltrušaitis et al., 2019]. Adopting the late fusion strategy has resulted in a more effective and robust performance in both multimodal [Alqhtani et al., 2018, Snoek et al., 2005] and monomodal applications [Blei et al., 2003, Gao et al., 2015, Wu et al., 2004]. While the improved performance can be application dependent [Balazs and Velásquez, 2016, Snoek et al., 2005], the reason for its robustness appears to be related to the consideration of the semantic information of features in late fusion [Snoek et al., 2005], even in monomodality [Blei et al., 2003, Gao et al., 2015]. However, late fusion can make learning correlations between multimodel features less effective [Baltrušaitis et al., 2019, Snoek et al., 2005] because these learned features are no longer flexible and they do not necessarily resemble their original data sources.

3.2.3.3 Hybrid Fusion

On the basis that no specific fusion approach always performs best, a combination scheme is necessary. A hybrid strategy in multimodal fusion includes performing fusion at both the feature (i.e., early) and decision (i.e., late) levels to solve problems in multimodal data analysis [Atrey et al., 2010, Wu et al., 2005]. In this thesis, and for relevant feature discovery, a similar approach is possible by combining different high-level features and low-level terms to exploit the advantages of the previously defined early and late fusion approaches. Therefore, we can define the hybrid fusion of text features as follows:

Definition 9 (Hybrid Fusion) *A fusion strategy that integrates early- and late-fused features* *into a composite semantic space between them and relevant and/or irrelevant documents.*

Existing hybrid fusion–based TFS models are the pattern–terms PDS [Zhong et al., 2012], MP [Yan et al., 2005], RDF$_1$ [Li et al., 2010] and RDF$_2$ [Li et al., 2015] models. In these models, fusing the semantic information of patterns with the richness (i.e., statistical properties) of low-level terms has led to a better performance [Wu et al., 2006, Zhong et al., 2012] than the late fusion of patterns only or the early fusion of terms only [Albathan et al., 2012, Wu et al., 2004]. Further, the hybrid fusion of patterns and terms facilitates a better exploitation of pattern mining in text, which was a challenging issue in only pattern-based TFS models [Li et al., 2015, 2011, Wu et al., 2004, Zhong et al., 2012].

3.2.4 Random Sets

A random set is a random object that has values as subsets taken from some space [Goutsias et al., 1997, Molchanov, 2005]. As a general mathematical modelling tool with many applications, random sets work as an effective measure of uncertainty in imprecise data for decision analysis [Nguyen, 2008]. For example, if Z and Ω are finite sets and Z is the evidence space, we propose

$$\Gamma : \ Z \to 2^{\Omega}$$

as a set-valued mapping from Z onto Ω. Because the SIF model aims to estimate a more accurate weight for topical terms, a probability function needs to be defined on the evidence space to specify the significance of the relationship that is governed by the set-valued mapping. Therefore, if P is a probability function that is defined on Z, the pair (P, Γ) is called a random set [Goutsias et al., 1997, Kruse et al., 1991]. However, because LDA defines a topic as a probability distribution over all terms in the collection, the random set will be extended to model the resultant complex relationships between topics and terms and vice versa, as described in the next section.

3.3 The Proposed SIF Model

The proposed SIF model (see Figure 3.1) fuses high-level topics with low-level terms to generalise the local weight of a topical term t in D^+, based on the set of topics Z that are generated from the paragraphs in G by exploring all possible relationships between different entities that influence the term-weighting process. The targeted entities in our model are paragraphs, topics and terms. The possible relationships between these entities are complex (a set of one-to-many or many-to-one relationships). For example, a paragraph can have multiple topics, whereby each topic is a probability distribution over all terms in Ω. Inversely, a topic can be discussed in many paragraphs, and a term can frequently appear in many topics and paragraphs.

An experimental practice commonly employed by most popular TFS models is to select top-k terms from Ω before training that is based on weighting schemes [Li et al., 2015, 2010, Zhong et al., 2012]. In the case of LDA, some top-k terms from each topic are also used for many applications [Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2014b, 2015]. However, as LDA topics are a mix of multiple probability distributions of terms, we argue that in TFS for relevance discovery, such a practice can lead to the loss of some relevant features, particularly those that are less frequent in the collection. This practice can also make the TFS model sensitive to the adopted weighting scheme. Therefore, rather than pre-selecting top-k terms from either Ω or each $z_i \in Z$ in the collection, our SIF model extends multiple random sets for this task.

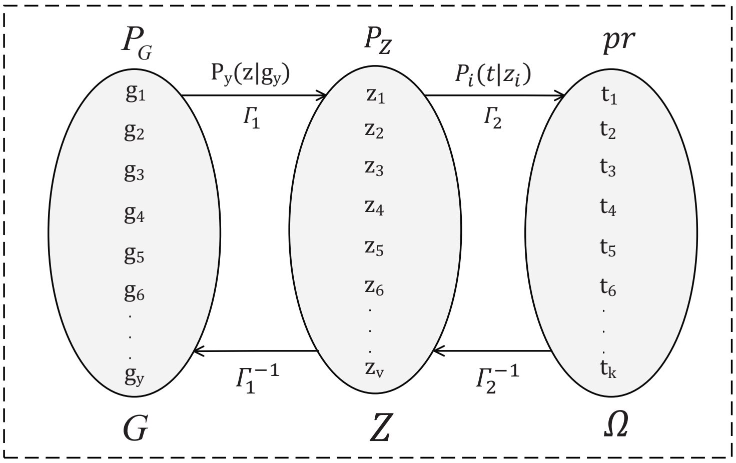


Figure 3.2: The feature fusion module of the SIF model and the mapping of Γ and Γ^{-1}.

In this model, two ERSs and their inverses are proposed to describe such complex relationships, in which each ERS can be interpreted as a probability function from which the importance of the main entity in the relationship can be determined. The proposed ERS theory is then used to develop a new weighting function to generalise LDA's local term probability to a global one that is locally descriptive, as the relevance details of the term in each document in the collection are considered. The generalised term weight is also more discriminating, as it accurately represents the relevance of the term to the user information needs, especially when combined with the global document frequency. Figure 3.2 shows the structure of the SIF model's feature fusion module, as well as the adopted entities and proposed ERSs, including their inverses. The details of the ERSs are described in the next section.

3.3.1 Extended Random Sets

Let $D^+ = \{d_1, d_2, d_3, \ldots, d_m\}$ be a set of M relevant long documents. Each document d_x consists of S paragraphs, such as $d_x = \{g_1, g_2, g_3, \ldots, g_s\}$. A paragraph g_y is a bag of terms; for example, $g_y = \{t_1, t_2, t_3, \ldots, t_k\}$. Assuming we have a set of latent topics $Z = \{z_1, z_2, z_3, \ldots, z_v\}$ that are extracted by the LDA from G as the set of all paragraphs in D^+, a topic z can be defined as a probability distribution over the set of terms Ω, in which $terms(g_y) \subseteq \Omega$ for every paragraph $g_y \in G$.

However, as a term t can appear in multiple topics, there is a need to estimate the topical significance of the term t by measuring the strength of its relationship with each topic $z_i \in Z$. Therefore, we extend Γ to an extended set-valued mapping [Li, 2003] as follows:

$$\psi :: Z \to 2^{\Omega \times [0,1]}$$

which satisfies

$$\sum_{(t,p) \in \psi(z)} p = 1$$

for each $z \in Z$, where Z is a set of topics (or evidences) and Ω is a set of terms (objects) as previously defined.

3.3.2 Generalised Weighting Scheme

Let P be a probability function on Z, such that

$$\sum_{z \in Z} P(z) = 1$$

We call (ψ, P) an ERS. For each $z_i \in Z$, let $P_i(t|z_i)$ be a conditional probability function on Ω, such that

$$\Gamma(z_i) = \{t | t \in \Omega, P_i(t|z_i) > 0\}$$

while the inverse mapping of Γ is defined as follows:

$$\Gamma^{-1} :\ \Omega \rightarrow 2^{Z}$$

$$\Gamma^{-1}(t) = \{z \in Z | t \in \Gamma(z)\}$$

A probability function $pr(t)$ can be decided by the extended Γ on Ω, which satisfies

$$pr :: \Omega \rightarrow [0, 1]$$

as follows:

$$pr(t) = \sum_{z_i \in \Gamma^{-1}(t)} \Big(P(z_i) \times P_i(t|z_i) \Big) \tag{3.1}$$

where $pr(t)$ is the generalised weight of term t at the collection level that LDA does not estimate, $P(z_i)$ represents the weight of topic z_i, $P_i(t|z_i)$ as the probability of term t in topic z_i, and Γ^{-1} is a mapping function.

Similarly, as a topic z_i might appear in multiple paragraphs, it is necessary to estimate its significance over G. Therefore, the extended random set Γ_1 is proposed to describe the relationships between paragraphs and topics by using the conditional probability function $P_y(z|g_y)$ as follows:

$$\Gamma_1 :\ G \rightarrow 2^{Z \times [0,1]}$$

$$\Gamma_1(g_y) = \{(z_1, P_y(z_1|g_y)), \ldots\}$$

Similarly, Γ_2 is also proposed to describe the relationship between topics and terms by using the defined conditional probability function $P_i(t|z_i)$ as follows:

$$\Gamma_2 :\ Z \rightarrow 2^{\Omega \times [0,1]}$$

$$\Gamma_2(z_i) = \{(t_1, P_i(t_1|z_i)), \ldots\}$$

Based on the previously defined inverse mapping, the inverse ERSs Γ_1^{-1} and Γ_2^{-1} are proposed. Γ_1^{-1} describes the inverse relationships between topics and paragraphs by using the probability function $P_z(z_i)$, such that

$$\Gamma_1^{-1} :\ Z \rightarrow 2^G$$

$$\Gamma_1^{-1}(z) = \{g_y | z \in \Gamma_1(g_y)\}$$

Conversely, Γ_2^{-1} describes the inverse relationships between terms and topics by using the probability function $pr(t)$, such that

$$\Gamma_2^{-1} :\ \Omega \rightarrow 2^Z$$

$$\Gamma_2^{-1}(t) = \{z | t \in \Gamma_2(z)\}$$

3.3.2.1 Generalising Topic Weight

To estimate the generalised term t weight in collection D^+, we need to estimate two probabilities that are based on Γ_1^{-1} and Γ_2^{-1}. The first is topic weight, which is the probability of each topic $P_z(z_i)$ in each paragraph in G in which we assume $P_G(g_y) = \frac{1}{N}$, where N is the total number of paragraphs in G as follows:

$$P_z(z_i) = \sum_{g_y \in \Gamma_1^{-1}(z_i)} \Big(P_G(g_y) \times P_y(z_i | g_y)\Big)$$
$$= \frac{1}{N} \sum_{g_y \in \Gamma_1^{-1}(z_i)} P_y(z_i | g_y) \tag{3.2}$$

where $P_y(z_i|g_y)$ is estimated by LDA, g_y refers to the y^{th} paragraph in G and Γ_1^{-1} is a previously defined mapping function.

3.3.2.2 Generalising Topical Term Weight

Second, for each topic z_i in Z, we must estimate the conditional probability of term t, given topic z_i, $P_i(t|z_i)$. The generalised term weight can thus be calculated using Equation 3.1, which can be expanded by using Equation 3.2 as follows:

$$
\begin{aligned}
pr(t) &= \sum_{z_i \in \Gamma_2^{-1}(t)} \Big(P_z(z_i) \times P_i(t|z_i) \Big) \\
&= \sum_{z_i \in \Gamma_2^{-1}(t)} \left[\left(\frac{1}{N} \sum_{g_y \in \Gamma_1{}^{-1}(z_i)} P_y(z_i|g_y) \right) \times P_i(t|z_i) \right] \\
&= \frac{1}{N} \sum_{z_i \in \Gamma_2^{-1}(t)} \left[P_i(t|z_i) \times \left(\sum_{g_y \in \Gamma_1^{-1}(z_i)} P_y(z_i|g_y) \right) \right]
\end{aligned}
\tag{3.3}
$$

3.3.3 Score Fusion Scheme

Finally, the global term score $s(t)$ at the collection level is calculated as follows:

$$s(t) = pr(t) \times df(t) \tag{3.4}$$

where $pr(t)$ is the generalised weight of term t, which is estimated previously by Equation 3.3, and $df(t)$ is the document frequency of term t.

3.3.4 Hybrid Fusion Algorithm

Algorithm 1 describes our SIF model in which the term weighting function (Equation 3.3) is its core. The algorithm begins with an initialisation step for all terms in Ω (steps 2–3). Then, the algorithm splits all paragraphs in the training documents D^+ (steps 5–7) after removing stop words and stemming all terms in each paragraph. Then, the algorithm uses LDA to generate two representations (steps 9–10). The first representation is the paragraph–topics coverage (i.e., paragraph–topic distributions). The second representation is a specified number of latent topics (10 topics in our case) ($V = 10$), which are generated from the set of paragraphs G. While $V = 10$ is reported in [Gao et al., 2015] as the best value for the used 50 collections of the RCV1 dataset, our SIF model tends to be insensitive to the hyperparameter V (see Section 6.9.1).

Then, the algorithm calculates the term weight based on Equation 3.3 for each term in Ω (steps 12–22). To do so, the algorithm first applies Equation 3.2 to calculate the topic probability $P_z(z_j)$ for each topic $z_j \in Z$ in all paragraphs in G (steps 13–16). Then, the algorithm continues to calculate the term probability in topics that contain the same term (steps 17–21) based on

Equation 3.3. The previous steps generalise the local term weight to a global one ($pr(t_i)$). Step 22 combines both global weights ($pr(t_i)$ and the document frequency $df(t_i)$). Notably, paragraph splitting, stop word removal, term stemming and LDA topic extraction can be done once and offline in this model.

3.3.4.1 Time Complexity Analysis

The proposed SIF model is trained offline using a small set of relevant documents. The experimental results in Section 6.8.1 show that SIF outperformed all baseline models in both IF and RRT tasks. The experiments demonstrated that SIF does not need a large training set. On average, our model needs only 13 relevant documents to perform effectively. However, SIF's efficiency depends largely on LDA's time complexity, which tends to be affected by the Gibbs sampling algorithm in which each of its iterations increases linearly with the number of topics V and number of documents (i.e., the number of paragraphs N in our case). Thus, as in [Gao et al., 2015, Wei and Croft, 2006], the LDA time complexity can be estimated as $\mathcal{O}(V \times N)$.

However, as our SIF model is trained offline, it only needs LDA to be run once. Also, as SIF is not sensitive to V, the time complexity of LDA is proportional to $\mathcal{O}(N)$. By analysing the time complexity of Algorithm 1—especially the core section (steps 12–22)—we can see that lines two to 22 take $\mathcal{O}(K \times V \times N)$ basic operations to complete, where K is the size of the vocabulary Ω, V is the number of topics and N is the number of paragraphs in the collection. Because the number of topics can be as few as one and SIF performance is not sensitive to it, the required time complexity is practically estimated to be $\mathcal{O}(K \times N)$.

3.4 Chapter Summary

This chapter described SIF, an innovative and effective fusion-based TFS model for relevance discovery. SIF extends multiple random sets to model the imprecise and complicated relationships between terms, topics and paragraphs to effectively manage the hybrid fusion of different lexical and topical features from a collection of relevant documents. Based on the proposed ERS theory, a score fusion scheme is developed to estimate the relevance of topical terms at the collection level. The estimated score accurately reflects the relevance of these terms to the user information needs and maintains the same semantic meaning of the terms across all relevant documents. The proposed model demonstrates the effectiveness of the hybrid fusion strategy, using high-level topical features and low-level terms statistics in an unsupervised

Algorithm 1: Hybrid fusion-based TFS algorithm

```
Input  : A set of relevant documents $D^+$, the vocabulary list $\Omega$ and total number of
         topics $V$
Output: A function $s : \Omega \rightarrow [0, \mathbb{R})$
$Z = T = G = \emptyset$;
foreach $t_i \in \Omega$ do
    $pr(t_i) = 0$;
// split all paragraphs in $D^+$
foreach $d_x \in D^+$ do
    foreach $g_y \in d_x$ do
        $G = G \cup \{g_y\}$;
$N = |G|$;
Generate paragraph-topic proportions $(\vartheta_{y,1}, \ldots, \vartheta_{y,v})$ by applying LDA to $G$;
Generate topics $Z = \{z_1, \ldots, z_v\}$ by applying LDA to $G$;
// calculate $t_i$ fused score based on Eq 3.3 & Eq 3.4
foreach $t_i \in \Omega$ do
    foreach $z_j \in Z$ do
        $P_{z_j} = 0$;
        foreach $g_y \in G$ do
            $P_{z_j} = P_{z_j} + \vartheta_{y,j}$;
        if $t_i \in z_j$ then
            $t' = \left( \frac{tf(t_i, z_j)}{\sum_{t \in z_j} tf(t)} \right) \times P_{z_j}$;
        else
            $t' = 0$;
        $pr(t_i) = pr(t_i) + t'$;
    $s(t_i) = \frac{pr(t_i) \times df(t_i)}{N}$;
```

learning setting. Based on the first 50 assessors' collections of the standard RCV1 dataset, TREC relevance judgements and seven performance measures, the experimental results (see Section 6.8.1) showed that the proposed SIF model significantly outperformed many state-of-the-art baseline models for IF and RRT in all evaluation metrics, despite the fusion strategies used or type of text features adopted.

However, the proposed SIF model is not without limitations. It assumes that a topical term is equally important to each relevant document. This assumption can be too simple because it ignores the local details about the term in each relevant document, knowing that the term is more likely to appear unevenly across the paragraphs of relevant documents. Thus, it is essential to revisit SIF and re-estimate the term's importance at the document level before it is globally generalised at the collection level. The next chapter describes SIF2, a more effective TFS model that overcomes the limitations of the SIF model.

Moreover, many existing TFS techniques for relevance discovery have no mechanism to consider the evidence of relevance in a relevant document. These TFS techniques assume either all paragraphs in the relevant document are equally relevant or all information in the document is necessary. A document is commonly labelled as relevant because it contains a small part(s) of relevant information in its segments; however, the non-relevant parts can introduce uncertainties into the discovered relevant features. Therefore, in the next chapter, we will also introduce the UR method, which very effectively deals with uncertainties in relevant features discovered by most existing TFS models and frameworks.

Chapter 4

Dealing with Uncertainties in Relevant Features

This chapter introduces two novel and highly effective fusion-based models for handling uncertainties in relevant features that were discovered in a collection of documents discussing user information needs. The first model is called SIF2 and is proposed to overcome the limitation of our SIF model presented in Chapter 3 and the limitations of similar TFS models. The proposed SIF2 model delves into each document's details to estimate more accurate weights for topical terms before generalising and integrating them with globally informative statistics. The second model is developed to tackle the uncertainties problem that arises from irrelevant parts of relevant documents. The UR method is proposed to incorporate paragraph-relevance evidence estimated from document paragraphs and use these pieces of evidence to revise relevant features discovered through various existing TFS models and frameworks. The details of the SIF2 model and the UR method are presented in Section 4.1 and Section 4.2, respectively, and the summary of this chapter is presented in Section 4.3. An extensive experimental evaluation of the SIF2 and UR models for IF and RRT is presented in Chapter 6.

4.1 The Proposed SIF2 Model

4.1.1 Introduction

Most fusion-based TFS models, including SIF, estimate the relevance of features that describe user information needs globally at the collection level, assuming that these features are equally important to each document in the collection [Greiff, 1998, Robertson and Zaragoza, 2009, Shirakawa et al., 2015]. In this study, we argue that such an assumption is too simple given that long documents can discuss many unbalanced topics across their paragraphs, and even these paragraphs can randomly describe multiple specific themes [Alharbi et al., 2017c, Anastasiu et al., 2013, Chien, 2016, Gao et al., 2015]. Therefore, these TFS models must consider the relevant details of features locally within each document before generalising their relevance to all user's needs in the collection. However, as noted previously, most fusion-based models were not developed on the basis that long documents or user information needs can exhibit multiple topics; this negatively affected their performances in selecting relevant features [Alharbi et al., 2017c, Bashar et al., 2016, Gao et al., 2017, 2015].

Topic-based models are explicitly developed presuming that documents contain multiple topics [Blei et al., 2003, Hofmann, 2001]. LDA is the most popular statistical topic modelling algorithm and has many applications, including relevant feature discovery [Alharbi et al., 2017c, Bashar et al., 2016, Gao et al., 2015]. However, LDA estimates terms' relevance weights on a document-by-document basis using the local topics–document probability proportions and the global term–topics assignments [Blei et al., 2003, Gao et al., 2015]. It does not automatically consider the sub-hierarchal features of the document, such as its paragraphs–topics distributions or the features higher up in the hierarchy that represent the full collection. Also, LDA represents each generated topic as a probability distribution over all terms in the collection [Bashar and Li, 2017, Blei et al., 2003]. Such global representation might not accurately reflect the local relevance of topical terms at the document level because these terms are not equally distributed over all documents in the collection. Therefore, term weights assigned by the LDA term probability function do not accurately reflect the importance of these terms in their local documents or the collection. Recent studies in [Alharbi et al., 2018a, 2017b, Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2015] confirmed that the LDA probability function negatively influenced the LDA's performance in discovering relevant features.

Relevant terms can be identified in a specific collection by fusing various instances (i.e., evidence) of these terms in different representations [Croft, 2000, Zhang and Balog, 2017]. At the collection level, terms' global statistics, such as document frequency (df), are important pieces of evidence that represent terms more discriminatively [Man et al., 2009, Sebastiani, 2002]. Nevertheless, in IR, representing the relevance of terms using global weighting schemes cannot provide better retrieval results, because term global statistics cannot reveal the term's local importance at the document level [Macdonald and Ounis, 2010, Maxwell and Croft, 2013], and neither can the LDA. This research asked if there is a method to fuse the LDA's hierarchal features with informative collection statistical features (particularly pf and df) to overcome their limitations in representing the local and global relevance of terms to the user information preferences.

This study aimed to develop an effective, fusion-based TFS model called SIF2.[1] The model adopts a complex hierarchal representation for the collection, consisting of its documents, paragraphs, latent topics and all terms in the collection. Figure 4.1 illustrates the main elements of this representation and the different lexical and statistical features extracted from these entities to be fused by the feature fusion module. As in SIF, the feature fusion module of SIF2 is the main component of the model. This component models the complicated and imprecise relationships between these hierarchal entities and the different features, using multiple ERSs to estimate a relevance score fusion function.

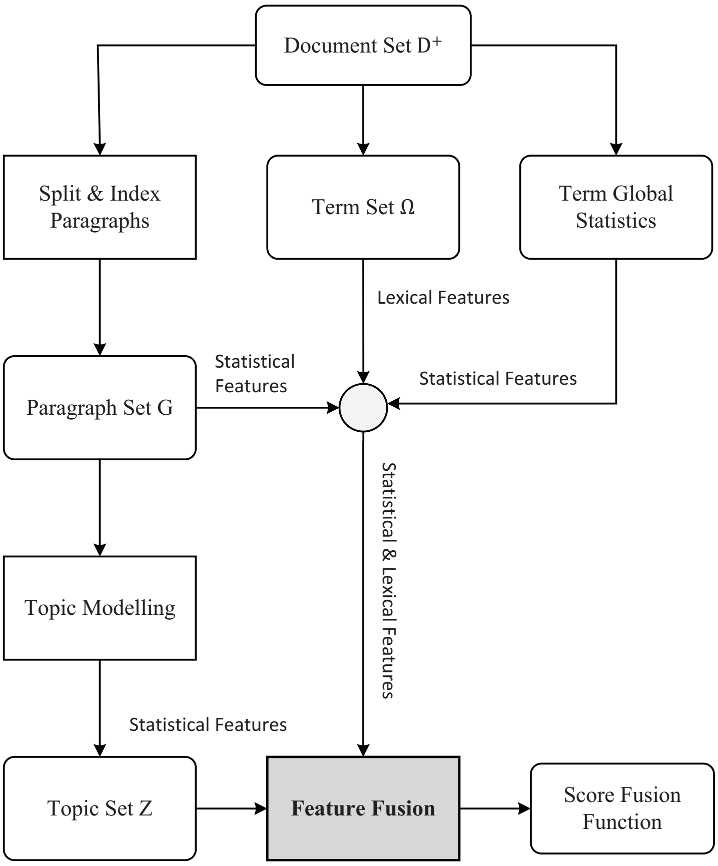


Figure 4.1: The SIF2 model structure.

The SIF2 model provides an elegant hybrid fusion approach that combines both high-level topics and local and global statistics of low-level terms to accurately score relevant topical terms at the collection level. This fused score effectively reflects the informativeness of the terms to the key topic of interest in a specific collection that describes user information needs.

[1] Parts of this model were published in [Alharbi et al., 2017a] and [Alharbi et al., 2018b]. The acronym '**SIF**' stands for the **S**election of **I**nformative **F**eatures while '**2**' refers to the use of both local and global statistics.

The experimental results presented in Section 6.8.2 demonstrate that the framework was highly effective and significantly outperformed both state-of-the-art and popular TFS methods in IF and RRT, regardless of the fusion technique or the type of text features they used. Further details about the proposed SIF2 model are described in the following sections. A background overview and some basic definitions are provided in Section 4.1.2, the theoretical details of the developed ERSs and the feature fusion functions are presented in Sections 4.1.3, 4.1.4 and 4.1.5 and the SIF2 algorithm and its time complexity analysis are described in Section 4.1.6.

4.1.2 Background and Basic Definitions

Assume that a researcher maintains a collection of long documents, denoted as D^+, describing a particular topic of interest that might also have multiple sub-topics or themes. For purposes of further investigation, the researcher wants to enrich the collection by collecting documents from the web. To achieve this goal, the researcher needs a model that can select and accurately weight terms to effectively describe the collection. The weighted terms [2] are used to gather relevant documents.

We assumed that the collection $D^+ = \{d_1, d_2, d_3, \ldots, d_x\}$ has M documents that are related to a particular topic of interest, which, as noted before, is different from a latent topic. A document d_x consists of a set of paragraphs S while a paragraph g consists of a bag of words and g_{xy} refers to the y^{th} paragraph of the x^{th} document. Therefore, the set of all paragraphs in the corpus is $G = \cup_{d_x \in D^+}\{g_{xy}|g_{xy} \in d_x\}$ and $S \subseteq G$. The set of all unique words in D^+ is $\Omega = \{t_1, t_2, t_3, \ldots, t_k\}$, where $K = |\Omega|$. SIF2 uses the LDA to discover a set of latent topics Z from G where V denotes the number of topics. The LDA is an effective model to discover hidden topics from a corpus, but it does not demonstrate sufficient performance in TFS for relevance discovery.

As noted before, LDA describes a topic $z_j \in Z$ as a probability distribution over all words in Ω using $p(t_i|z_j)$, in which $\sum_i^{|\Omega|} p(t_i|z_j) = 1$, where $1 \leq j \leq V$ and $t_i \in \Omega$. Also, LDA describes a document d_x by a probabilistic mixture of topics using $p(z_j|d_x)$. All hidden variables, $p(t_i|z_j)$ and $p(z_j|d_x)$, are inferred statistically by the Gibbs sampler [Steyvers and Griffiths, 2007]. Consequently, and based on Z, the local weight (i.e., probability) of word t_i in a document d_x can be estimated as $p(t_i|d_x) = \sum_{j=1}^{V} p(t_i|z_j) \times p(z_j|d_x)$. Therefore, for every

[2] In this chapter, we continue to use ’terms’, ’words’, ’keywords’ and sometimes ’features’ interchangeably unless explicitly stated otherwise.

topic $z_j \in Z$, estimating $p(t_i|d_x)$ requires the fusion of two hierarchal features: the word–topic assignment $p(t_i|z_j)$ and the topic–document distribution $p(z_j|d_x)$. However, we argue that using these features makes the LDA ineffective for selecting relevant terms in a specific collection, whether at document level or paragraph level (see LDA's experimental results in Section 6.8.2).

Therefore, adapting LDA to estimate words' informativeness has two challenges: a) how to localise global features for a more accurate estimation of their local relevance and b) how to fuse other hierarchal features for a better relevance estimation for topical terms. In the following section, we define some informative local features that will be used to represent relevance information in documents. These features will also be integrated by the SIF2 model to estimate the relevance of topical terms in the collection that describe user information needs.

4.1.2.1 Informative Text Features

As in the SIF model, the proposed SIF2 model adheres to the hybrid fusion strategy defined in Section 3.2.3.3 through integration between high-level topics and low-level terms. However, unlike SIF, the SIF2 model will estimate the relevance of each topical term at the document level, assuming that they have different degrees of relevance in each long document in the collection. Therefore, some informative topical and low-level statistical features are adopted and defined in this study as local features.

Local Features

Local features can be used to measure the importance of terms within a specific document or even a fixed-size window of text [Macdonald and Ounis, 2010, Maxwell and Croft, 2013, Pickens and Golovchinsky, 2008]. In low-level terms, popular local features are the local statistics of terms, such as term frequency tf, paragraph frequency pf and sentence frequency sf. Instead of using sf, which is comparable to tf, this study uses pf at the document level as a local low-level feature. The pf demonstrates better results than tf in representing the informativeness of relevant topical term in our SIF model, as can be observed in Table 6.23. However, to calculate $pf(t)$ at the document level, we update its previous definition in Section 3.2.2.1 to the following:

Definition 10 (Document–Paragraph Frequency) *The pf of a term t in document d_x, denoted as $pf(t, d_x)$, is the number of paragraphs in d_x that contain the term t. Thus, knowing that S is the set of all paragraphs in d_x, $pf(t, d_x)$ can be calculated as follows:*

$$pf(t, d_x) = \sum_{y=1}^{|S|} f_{g_y}(t)$$

where $|S|$ is the total number of paragraphs in d_x and $f_{g_y}(t)$ is defined as follows:

$$f_{g_y}(t) = \begin{cases} 1, & \textit{if } t \in g_y \\ 0, & \textit{otherwise} \end{cases}$$

Since low-level terms do not contain any semantic information and do not assume that a document can exhibit multiple topics, using their local statistics alone is not sufficient for estimating the relevant information in the document. Therefore, a local feature that can reveal the topical coverage of the document is required. The LDA is used for this purpose to estimate what we call in this study a topical paragraph (tp) that will be used to estimate the topical coverage of the document. The topical paragraph is defined as follows:

Definition 11 (Topical Paragraph) *The tp of a paragraph g_y of document d_x, denoted as $tp(g_{xy})$, is the proportions of the probability distribution of g_{xy} over a specific number of latent topics as follows:*

$$tp(g_{xy}) = (\vartheta_{xy,1}, \vartheta_{xy,2}, \vartheta_{xy,3}, \ldots, \vartheta_{xy,V})$$

where $0 \leq \vartheta_{xy,V} \leq 1$ and V is the number of LDA topics.

Further, tp represents the topical coverage at the paragraph level, but in this study, we selected the document as our semantic space to estimate the relevance of topical terms, as noted previously. Therefore, and using the definition of tp, we call this semantic space a topical document (td) and define it as follows:

Definition 12 (Topical Document) *The td of document d_x, denoted as $td(d_x)$, is the sum of the proportions of the identical topics in every topical paragraph $tp(g_y)$ in d_x where $g_y \in d_x$. Therefore, the $td(d_x)$ can be calculated as follows:*

$$td(d_x) = \left(\sum_{g_y \in d_x}^{|S|} \vartheta_{y,1}, \ldots, \sum_{g_y \in d_x}^{|S|} \vartheta_{y,V} \right)$$

where $\vartheta_{y,i}$ is the proportion of topic i in paragraph y in document d_x, and $|S|$ is the total number of paragraphs in document d_x.

Figure 4.2 illustrates the informativeness of td in representing the topical coverage (i.e., topical information) of three different relevant documents taken from Collection 101 of the RCV1 dataset. The figure demonstrates that these documents almost cover the same topics of interest but with variant levels of significance. The figure experimentally justifies our claim that a topical term might have different degrees of relevance in each relevant document.

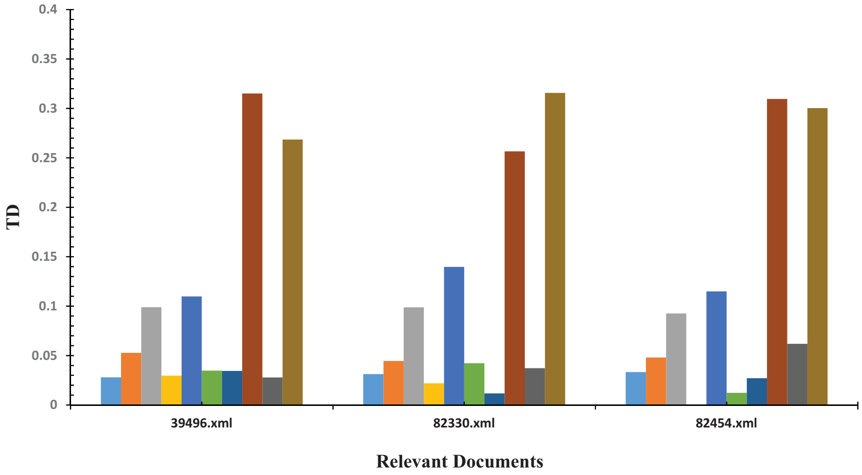


Figure 4.2: The td of three different relevant documents from Collection 101 of the RCV1 dataset using 10 LDA topics.

4.1.3 Extending Multiple Random Sets

Distinct hierarchal entities and their relationships to each other can affect the term scoring in a specific collection. As demonstrated in Figure 4.3, the SIF2 model uses four entities, which are the collection documents D^+, their paragraphs G, the LDA latent topics Z and the collection keywords Ω. Similar to our SIF model, SIF2 also models the complex relationships between these entities using the ERS theory to integrate and, thereby, generalise the weight of a local term to a global one that can be combined with a more informative global statistic.

However, in SIF2, we proposed three ERSs Γ_1, Γ_2 and Γ_3 and their inverses to model the one-to-many relationships between the used entities, as illustrated in Figure 4.3. In every ERS, including its inverse, a probabilistic function is used to describe a specific relationship and assign a weight that represents the strength of the relationship with the targeted entity. Then, a new score fusion function is developed by integrating the proposed ERSs. The function assigns highly informative scores to topical terms in the collection, representing their relevance to what the user needs.

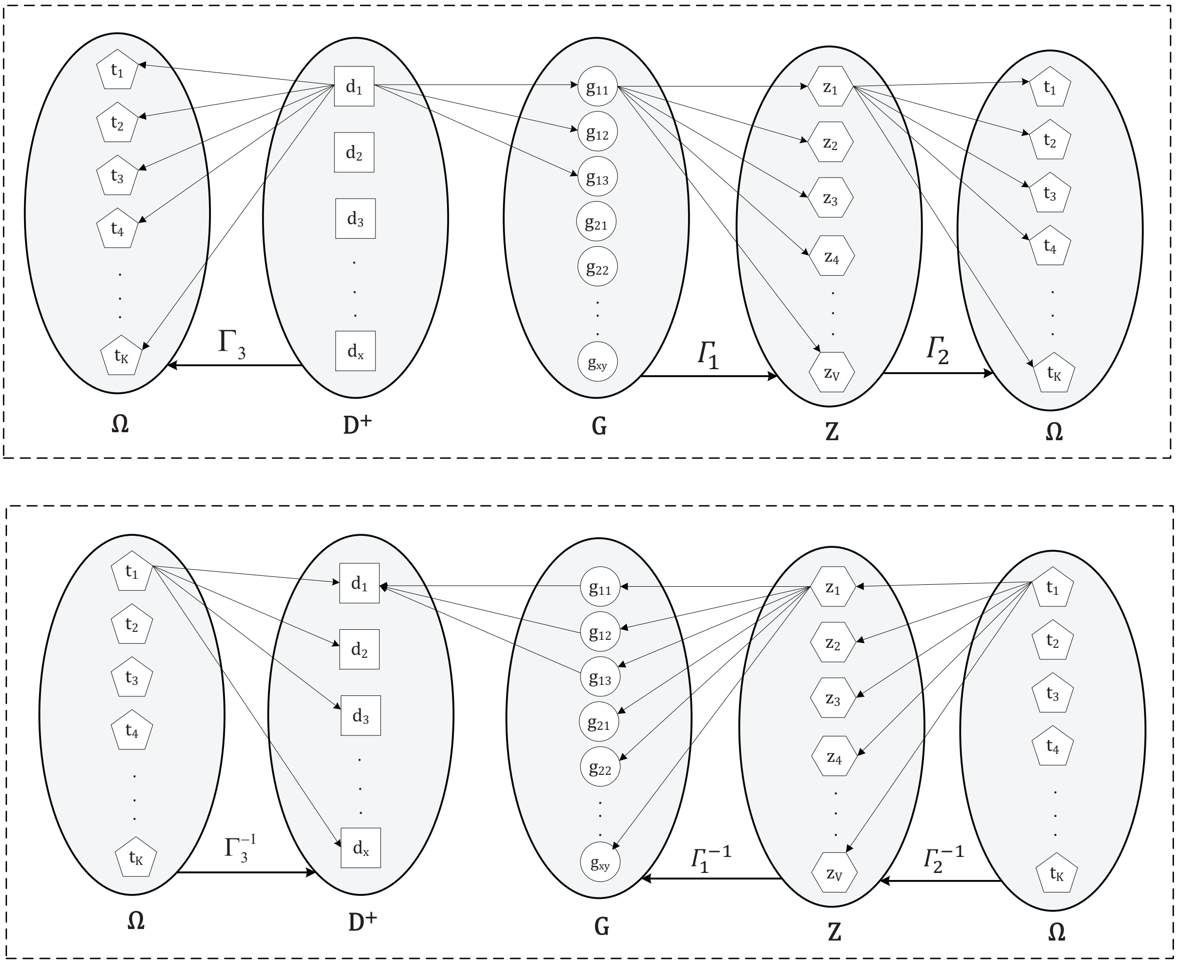


Figure 4.3: The feature fusion module of the SIF2 model and the mapping of Γ and Γ^{-1} between the used entities.

To effectively estimate the local relevance of topical terms in every document $d \in D^+$, we first must consider the hidden topics Z discussed in all documents in D^+ and their relationships with all terms in the collection (i.e., Ω). Therefore, we proposed the set-valued function

$$\Gamma : \ Z \to 2^{\Omega}$$

from Z onto Ω. However, it is important to estimate the strength of every relationship between a term and a topic in the collection. Therefore, let us consider Z the evidence space and propose

P as a probability function specified on Z. However, because LDA defines each topic as a probability distribution over all terms in Ω, not as a scalar value, and a term t can appear in many topics, we extended Γ as

$$\Psi :: Z \to 2^{\Omega \times [0,1]}$$

and it is called an extended set-valued mapping [Li, 2003] such that

$$\sum_{(t,p) \in \Psi(z)} p = 1$$

for each $z \in Z$.

However, because LDA assumes that a document contains multiple topics, then, for every topic $z \in Z$, we define a probability function P that satisfies

$$\sum_{z \in Z} P(z) = 1$$

Therefore, we can call the pair (Ψ, P) an ERS, as noted previously. Consequently, and for every $z_i \in Z$, we define $P_i(t|z_i)$ as a conditional probability function on the set of terms Ω to describe the new relationship between the term t and a set of topics such that the mapping

$$\Gamma(z_i) = \{t|t \in \Omega, P_i(t|z_i){>}0\}$$

However, as our ultimate goal is to estimate the relevance of t in a document d_j, not only in a topic z_i, we much first estimate t weights in all topics. Therefore, we consider Γ^{-1} the inverse function of Γ and define it as follows:

$$\Gamma^{-1} : \ \Omega \to 2^{Z}$$

$$\Gamma^{-1}(t) = \{z \in Z|t \in \Gamma(z)\}$$

Nevertheless, as noted earlier, LDA estimates Z from all terms in the collection. Knowing that these terms appear unevenly across all documents in the collection and that all topics in Z appear in an unbalanced way in each document, based on the definitions presented in the last section, we must localise all topical information to each document individually and the relevance of t at the document level, not the topic level. Therefore, based on the ERS defined above, we propose the score fusion function $sr_d(t)$ on Ω such that

$$sr_d :: \Omega \rightarrow \mathbb{R}$$

as follows:

$$sr_d(t) = \sum_{d_j \in \Gamma^{-1}(t)} \left\{ \frac{1}{P_j(t|d_j)} \cdot \left[\sum_{z_i \in \Gamma^{-1}(t)} \Big(P_z(z_i) \times P_i(t|z_i) \Big) \right] \right\} \tag{4.1}$$

where $sr_d(t)$ is the fused score of topical term t at the document level. The functions $P_j(t|d_j)$ and $P_i(t|z_i)$ calculate the conditional probability of the term t in document d_j and topic z_i, respectively. Finally, $P_z(z_i)$ estimates the generalised weight of topic z_i in d_j as it will appear in Equation 4.2.

4.1.4 Integrating Informative Features

To estimate $sr_d(t)$, we must investigate d_j and accurately measure the strength of all possible relationships between 1) S and Z, 2) Z and t, and 3) t and D^+. Therefore, the ERS Γ_1 defines the conditional probability function $P_{xy}(z|g_{xy})$ on the set of paragraphs G to describe the one-to-many relationship between a paragraph and a topic as

$$\Gamma_1 : \ G \rightarrow 2^{Z \times [0,1]}$$

$$\Gamma_1(g_{xy}) = \{(z_1, P_{xy}(z_1|g_{xy})), \ldots\}$$

Similarly, as a topic can have many terms, Γ_2 defines $P_i(t|z_i)$ on Z as another conditional probability function that estimates the probability of a term based on its appearance in each topic $z_i \in Z$ as

$$\Gamma_2 : \ Z \rightarrow 2^{\Omega \times [0,1]}$$

$$\Gamma_2(z_i) = \{(t_1, P_i(t_1|z_i)), \ldots\}$$

Further, Γ_3 is also proposed to describe the relationship between documents and terms using the defined probability function $P_j(t|d_j)$ as

$$\Gamma_3 : D^+ \rightarrow 2^{\Omega \times [0,1]}$$

$$\Gamma_3(d_j) = \{(t_1, P_j(t_1|d_j)), \ldots\}$$

Based on the inverse mapping described above, we have Γ_1^{-1}, Γ_2^{-1} and Γ_3^{-1}. The Γ_1^{-1} describes the inverse relationships between topics and paragraphs using the probability function $P_z(z_i)$, such that

$$\Gamma_1^{-1} : \; Z \rightarrow 2^G$$

$$\Gamma_1^{-1}(z) = \{g_{xy}|z \in \Gamma_1(g_{xy})\}$$

Γ_2^{-1}, conversely, describes the inverse relationships between terms and topics using the $P_i(t|z_i)$ function such that

$$\Gamma_2^{-1} : \; \Omega \rightarrow 2^Z$$

$$\Gamma_2^{-1}(t) = \{z|t \in \Gamma_2(z)\}$$

Γ_3^{-1} describes the inverse relationships between terms and documents using the function $P_j(t|d_j)$ such that

$$\Gamma_3^{-1} : \; \Omega \rightarrow 2^{D^+}$$

$$\Gamma_3^{-1}(t) = \{d_j|t \in \Gamma_3(d_j)\}$$

Inversely, as a topic also can appear in one or more paragraphs that belong to a certain document, Γ_1^{-1} is proposed to describe such a relationship using the $P_z(z_i)$ function, in which a subset of paragraphs S will only be mapped to its document as

$$\Gamma_1^{-1}(z) = \{g_{xy}|z \in \Gamma_1(g_{xy}), g_{xy} \in S\}$$

Similarly, as a term t in a specific document can occur in multiple topics, Γ_2^{-1} is also proposed to govern this relationship using the probability function $sr_d(t)$ as

$$\Gamma_2^{-1}(t) = \{z|t \in \Gamma_2(z)\}$$

4.1.4.1 Estimating Topical Relevance

To estimate the relevance of term t in a document d_x, Γ_1^{-1}, Γ_1^{-2} and Γ_3^{-1} are used to calculate two probabilistic scores based on the definitions of tp and td. The first score represents the topical relevance at the document level $P_z(z_i)$ for every topic that appears in paragraph $g_y \in d_x$. The Γ_1^{-1} is used to integrate the topic–paragraph distribution $P_{xy}(z_i|g_{xy})$ for estimating its topic–document marginal probability distribution. We assume $P_G(g_{xy}) = \frac{1}{N}$, denoting that every $g_y \in d_x$ is likely equally important, and $N = |S|$ as follows:

$$P_z(z_i) = \sum_{g_{xy}\in\Gamma_1^{-1}(z_i)} (P_G(g_{xy}) \times P_{xy}(z_i|g_{xy}))$$

$$(4.2)$$

$$= \frac{1}{N} \sum_{g_{xy}\in\Gamma_1^{-1}(z_i)} P_{xy}(z_i|g_{xy})$$

where $P_{xy}(z_i|g_{xy})$ is estimated based on the definition of tp, and g_{xy} denotes paragraph y of document x.

4.1.4.2 Estimating Term Relevance

The second score estimates the relevance of t at the document level and is calculated first using Γ_1^{-2} for every topic $z_i \in Z$ based on the conditional probability distribution $P_i(t|z_i)$. However, Γ_3^{-1} is adopted to localise the globally calculated probabilities based on $P_j(t, d_j)$. Therefore, the fused term t score at a document d level can be estimated by substituting $P_z(z_i)$ in Equation 4.1 with its formula in Equation 4.2 as follows:

$$sr_d(t) = \frac{1}{N} \sum_{d_j\in\Gamma_3^{-1}(t)} \left\{ \frac{1}{P_j(t|d_j)} \cdot \left[\sum_{z_i\in\Gamma_2^{-1}(t)} P_i(t|z_i) \times \left(\sum_{g_{xy}\in\Gamma_1^{-1}(z_i)} P_{xy}(z_i|g_{xy}) \right) \right] \right\} \quad (4.3)$$

4.1.5 Score Fusion Scheme

As noted previously, the proposed SIF2 model adopts the hybrid fusion strategy through the integration of high-level topical features and both local and global statistics of low-level terms. The proposed $sr_d(t)$ function estimated the local relevance of term t in a specific document though the fusion of informative local features. However, relevant terms that describe user information needs must be estimated at the collection level, not only at a specific document level . Therefore, we estimated the global score for a term t at the collection level, denoted as $sc(t)$, to be the sum of its $sr_d(t)$ in every document $d_i \in D^+$ integrated with the informative global statistic df, as in the SIF model. The $sc(t)$ is calculated as follows:

$$sc(t) = df(t) \cdot \sum_{t \in d_i, d_i \in D^+} sr_{d_i}(t) \tag{4.4}$$

where $df(t)$ is the document frequency of term t and $sr_{d_i}(t)$ is the fused score of t in document d_i.

4.1.6 Hybrid Fusion Algorithm

Algorithm 2 illustrates the details of the proposed SIF2 model, in which Equation 4.3 represents the main function in the model. The algorithm follows the same pre-processing and initialisation steps of the SIF algorithm (Algorithm 1), except that each paragraph in the collection is indexed to be mapped to its containing document. The LDA is also used with the SIF2 model to generate 10 topics from the set of all paragraphs G, as illustrated in steps 8–9. The number of LDA topic was set experimentally, but SIF2 is insensitive to this hyperparameter. Steps 11–25 are the core steps of the algorithm, based on the details of Equation 4.4.

4.1.6.1 Time Complexity Analysis

The proposed SIF2 models inherited the positive aspects of our SIF model in terms of the insensitivity to the number of LDA topics (i.e., V), as illustrated in Figure 6.23. Also, SIF2 does not require a large training set and is trained offline with a single LDA run. Due to SIF2's insensitivity to V, the time complexity of LDA is $\propto \mathcal{O}(N)$, where N is the total number of paragraphs in D^+. SIF2's algorithm resembles much of the contents of SIF's; the only noticeable difference is the use of a third loop to iterate through the number of relevant documents M in the collection. Therefore, the time complexity of SIF2's algorithms is $\propto \mathcal{O}(K \times M \times N)$, where K is the vocabulary size.

Algorithm 2: Score fusion scheme

```
Input  : A set of relevant documents $D^+$, the vocabulary $\Omega$ and total number of
         topics $V$
Output: A function $sc : \Omega \rightarrow [0, \mathbb{R})$
$Z = T = G = \emptyset$;
foreach $t_i \in \Omega$ do
    $sc(t_i) = 0$;
foreach $d_x \in D^+$ do
    foreach $g_y \in d_x$ do
        $G = G \cup \{g_{xy}\}$;
$N = |G|$;
Generate paragraph-topic proportions $(\vartheta_{xy,1}, \ldots, \vartheta_{xy,v})$ by applying LDA to $G$;
Generate topics $Z = \{z_1, \ldots, z_v\}$ by applying LDA to $G$;
// calculate $sc(t)$ based on Equation 4.4
foreach $t_i \in \Omega$ do
    foreach $d_x \in D^+$ do
        if $t_i \in d_x$ then
            $w = \left(\frac{pf(t_i,d_x)}{pf(t_i)}\right)$;
            foreach $z_j \in Z$ do
                $P_{z_j} = 0$;
                foreach $g_{xy} \in d_x$ do
                    $P_{z_j} = P_{z_j} + \vartheta_{j,xy}$;
                if $t_i \in z_j$ then
                    $w' = \left(\frac{tf(t_i,z_j)}{\sum_{t \in z_j} tf(t)}\right)$;
                    $t' = (w' \div w) \times P_{z_j}$;
                else
                    $t' = 0$;
                $sr_{d_x}(t_i) = sr_{d_x}(t_i) + t'$;
    $sc(t_i) = \frac{sr_{d_x}(t_i) \times df(t_i)}{N}$;
```

4.2 The Proposed UR Method

4.2.1 Introduction

As noted previously, relevance discovery algorithms face challenges in identifying relevant text features from both a theoretical and empirical viewpoint [Alharbi et al., 2017b, Li et al., 2015, 2010]. One main challenge is the uncertainties associated with the features discovered from irrelevant or less relevant paragraphs that might exist in relevant documents. This is because a document can be labelled relevant if only a small part of it contains relevant information, as previously illustrated in Figure 1.3. Using only document-level evidence can select features from all parts of the document, which can lead to uncertainties and scatter the focus of the selection algorithm because the features coming from irrelevant parts do not describe user information needs. Therefore, the relevance of the corresponding part should be considered when selecting features from it. Many studies have been conducted to develop TFS models of relevance discovery over the last few decades [Gao et al., 2015, Li et al., 2015, Man et al., 2009, Song et al., 2013, Tao et al., 2011]. However, most of these consider only the document- or collection-level evidence for discovering relevant features, which makes them vulnerable to the uncertainties present in a specific document or even the entire collection.

Research in IR has demonstrated that considering the evidence at the passage level can improve document retrieval accuracy, especially when documents are long or span different subject areas [Callan, 1994, Kozorovitsky and Kurland, 2011a, Liu and Croft, 2002]. Generally, the performance of IR models can dramatically improve depending on the amount of relevant evidence available in each passage [Anava et al., 2016, Callan, 1994, Fan et al., 2018]. Most existing IR research measures the amount of relevance between a fixed window-size passage and a user query through the estimation of some query similarity scores as the passage-level evidence [Bendersky and Kurland, 2010, Xi et al., 2001]. However, the explicit user query may not always be available, as in the case of IF, which forbids the estimation of such query similarity scores [Gao et al., 2015, Li et al., 2012]. Therefore, in a situation in which paragraphs are variant in size (i.e., no fixed window-size passages are considered), it becomes challenging to explicitly estimate paragraph-level relevance evidence in a set of relevant documents that describes user information needs. Also, it is equally difficult to use the estimated relevance at the paragraph level to reduce uncertainties of relevant features that already been discovered by existing TFS models and frameworks. Therefore, implicit mechanisms are required to estimate and then utilise paragraph-level relevance evidence.

Text feature fusion performed effectively in dealing with uncertainty through the combination of multiple evidences available in different high-level and low-level text features [Alharbi et al., 2018b, Gao et al., 2015, Li et al., 2015]. However, these features are more likely to be uncertain as they might be extracted from irrelevant or less relevant parts of documents. Therefore, it is challenging to know which features to fuse, how to deal with their inherent uncertainties, how to fuse them to estimate the relevance of a paragraph and, ultimately, how to use the paragraph-level evidence to deal with uncertainties in relevant features discovered by other relevance discovery models and frameworks. The latent topical features of LDA [Blei et al., 2003] seem to be better candidates for estimating the relevance available in different entities (e.g., document, paragraph or sentence) of a collection. This is because they are the only features explicitly generated based on the assumption that a text document (or even a paragraph) can discuss multiple topics or themes [Alharbi et al., 2017c, Gao et al., 2014b, 2015]. LDA defines each discovered topic as a multinomial distribution over the terms in the collection. It also represents each document or paragraph as a mixture of the discovered topics [Blei et al., 2003, Griffiths and Steyvers, 2004]. However, LDA, as an unsupervised learning algorithm, treats all documents or paragraphs equally and pays no attention to any relevance evidence that might be available in them.

Therefore, in this section, we describe the uncertainties reduction (UR) [3] method, which uses paragraph relevance to reduce the uncertainties of the relevant features discovered by existing models (e.g., BM25 [Robertson and Zaragoza, 2009], Rocchio [Rocchio, 1971], RFD$_2$ [Li et al., 2015]). The method adopts the late fusion strategy to integrate different features extracted from the relevance feedback collection as an implicit mechanism to estimate the paragraph relevance. We call the user information needs' specific subject matters 'topics'. For example, the user information needs of *global warming* may involve topics like *pollution*, *greenhouse gases* and *ozone layer depletion*. We assume that frequent topics in the relevance feedback collection are the most relevant ones and use them to estimate the relevance of paragraphs. LDA is used in this study to discover these topics from the collection. However, the UR method does not use topical terms (i.e., LDA term–topic distributions) to avoid any inherent uncertainties that might exist in these statistical features, knowing that they are estimated from all terms of all paragraphs in the collection without considering the relevance of any paragraph.

[3] An adapted version of this model was published in [Alharbi et al., 2018a].

A relevance feedback collection that discusses user information preferences, the relationships between distinct entities in the collection—namely, its documents, paragraphs, topics and terms—and the estimated strengths of these relationships can be modelled as extended set-valued observations [Alharbi et al., 2017b,c, 2018b]. The uncertainties in phenomena that can be observed and represented as multiple sets, not as exact points, can accurately be modelled using ERS [Li, 2003, Li and Zhong, 2003]. Therefore, in the UR method, multiple ERSs are developed to effectively model these complex relationships so that they can be understood and the uncertainties dealt with through the hybrid fusion of different representative features discovered from the selected entities. Based on the ERSs, a weight-scaling scheme is also developed to use the estimated paragraph-level relevance for uncertainties reduction. The scheme is applied to individual terms (i.e., lexical features), due to their flexibility and shareability between different entities and high-level features in the collection. Therefore, the developed scaling scheme is used to scale the weights of relevant term sets discovered by a TFS model as an implicit mechanism to reduce uncertainties in these terms. Then, the scaled relevant terms are re-ranked to represent a new term set that is less uncertain and more relevant to user information needs.

Figure 4.4 illustrates the structure of the UR method in which the feature fusion module is the main component. The figure also shows the related entities and the flow of different features from them to the feature fusion module. The UR's structure resembles that of SIF2 as the UR also estimates the relevance of a paragraph locally, at its document level, and globally, at the collection level. Section 6.8.3 in Chapter 6 presents the results of experiments conducted on the 50 human-assessed collections of documents from the standard RCV1 dataset and their TREC filtering topics, showing that the proposed UR method is highly effective in reducing uncertainties. When applied to the suitable existing TFS model, the improved model significantly outperforms all the other models in all evaluation metrics, regardless of the relevance discovery technique or the type of text features they use. More details about the proposed UR method are presented as follows: the problem formulation is introduced in Section 4.2.2, the relevance estimation of paragraphs and the developed scaling function are described in Section 4.2.3 and Section 4.2.4 describes how the estimated paragraphs' relevance can be used to reduce uncertainties in relevant features selected by any TFS model.

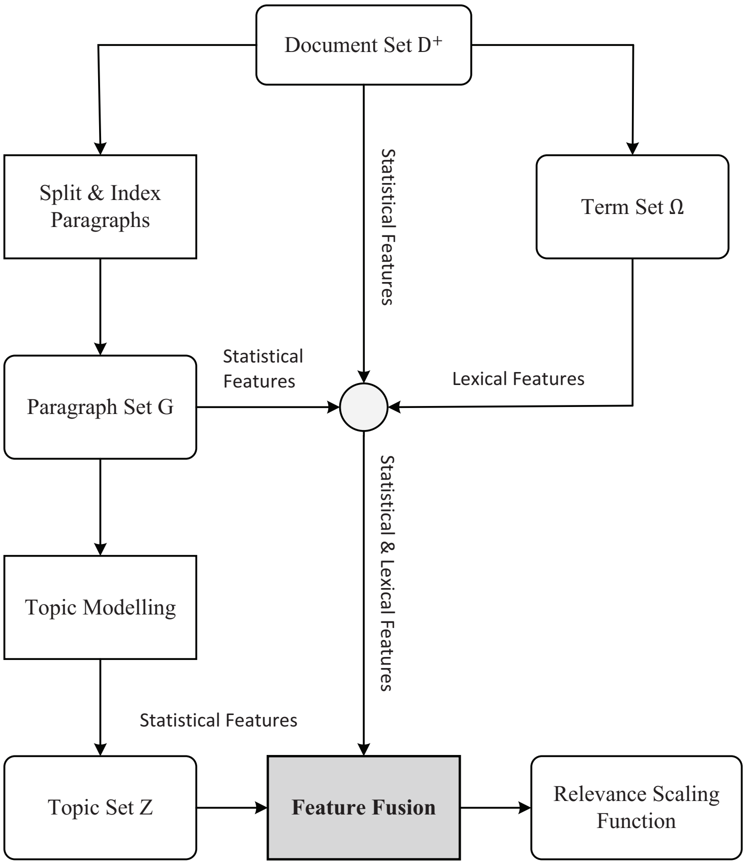


Figure 4.4: The UR method structure.

4.2.2 Problem Formulation

Given a set of documents D that discusses both relevant and irrelevant user information needs [Li et al., 2015, 2010], the set D^+ denotes the positive (i.e., relevant) documents in D such that $D^+ \subseteq D$, and D^- represents the set of negative (i.e., irrelevant) documents such that $D^- \subseteq D$, and, therefore, $D = D^+ \cup D^-$. A relevant long document $d_x \in D^+$ has a set of paragraphs S and the set G denotes all paragraphs in D^+, where g_{xy} is the y^{th} paragraph of the x^{th} document and $S \subseteq G$. Also, each paragraph is a bag of terms and Ω is the set of all terms in D^+. As each paragraph might discuss multiple sub-topics or themes, a set of statistical topics Z is extracted from G using the LDA model. These topics reduce the dimensionality of G to just a few topics, where V denotes the total number of topics in Z. The topics are integrated with other statistical features to estimate the relevance of each paragraph in G.

In this study, we assume that a paragraph g_{xy} has a local significance at its containing document and another global relevance significance at the D^+ collection. A long document can discuss many topics across its paragraphs, and the paragraphs can also exhibit multiple smaller themes [Blei et al., 2003, Gao et al., 2014b, 2015]. Therefore, a relevant paragraph should summarise this topical information described in its document. However, for user information needs, these topics and themes might be discussed randomly and unevenly across the relevant documents, which makes the local relevance estimation of the paragraph significantly unrepresentative of what the user needs. Therefore, a global relevance for the paragraph must also be estimated based on its local significances in all documents. However, as noted before, it is challenging to estimate paragraph relevance in the absence of a specific search guide for such relevance (e.g., a user query), knowing that paragraphs' terms can appear in many other paragraphs, documents and topics. The topics, also, can be randomly discussed in multiple documents and paragraphs.

Moreover, as LDA defines its topics as multiple probability distributions over all terms in Ω and represents each paragraph as a probabilistic mixture of all topics, it is difficult to model and understand the highly complex relationships between the entities that influence the relevance estimation of a paragraph, since they are not exact points. Therefore, as shown in Figure 4.5, multiple ERSs and their inverses are developed to model the complicated relationships between documents, their paragraphs, topics and terms. Further, a probability function is developed to estimate the strength of each relationship. Then, all functions are effectively combined to estimate the relevance of paragraphs based on their lexical and statistical features. More details about the proposed ERSs are provided in the next section.

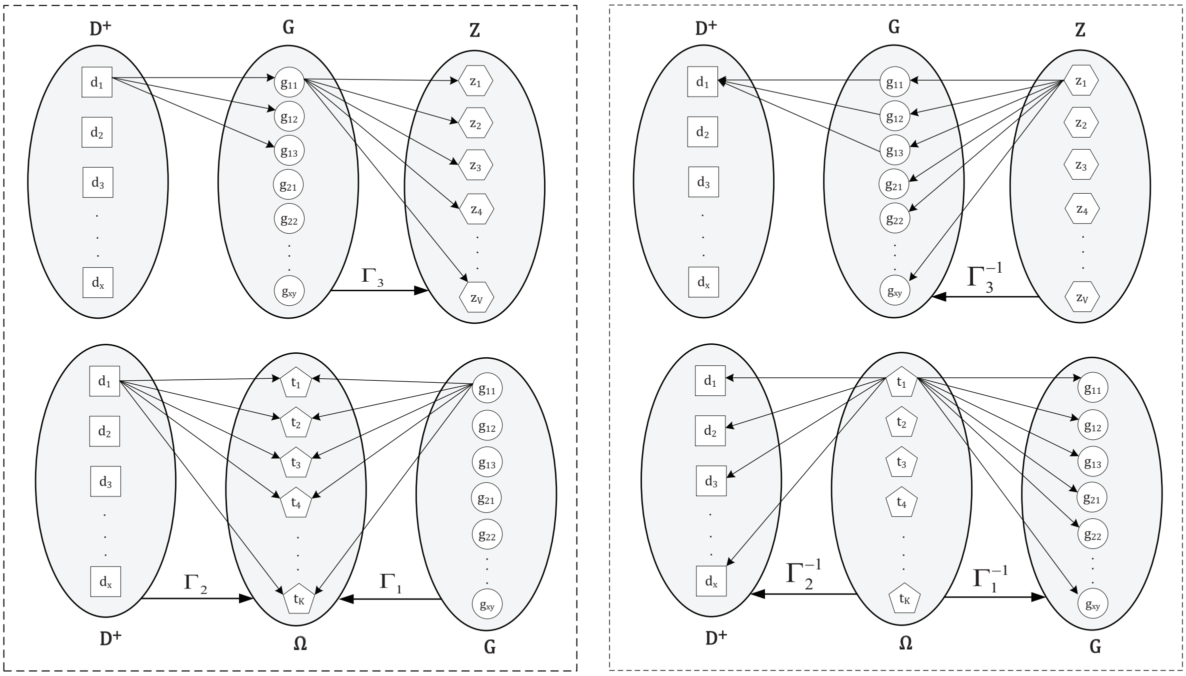


Figure 4.5: The feature fusion module of the UR method and the mappings of Γ (left) and Γ^{-1} (right).

4.2.3 Estimating Paragraph Relevance

To estimate the global relevance of a paragraph g_{xy} at the set D^+ level, first we measure the local significance of g_{xy} in its containing document (i.e., d_x) based on the relevance of the g_{xy} terms. However, many subsets of these terms can appear in many other paragraphs, documents and topics in the D^+ collection, and many of these topics can also be discussed in g_{xy}, knowing that each topic $z_j \in Z$ might also be exhibited in many paragraphs. Therefore, multiple probability distributions are defined and then modelled using multiple ERSs. Second, as in the SIF2 model, we assume that the global relevance of g_{xy} is the summation of its local relevance in each document $d_x \in D^+$. More details about the estimation of paragraph relevance are provided in the following two sections.

4.2.3.1 Local Relevance

As a paragraph is a set of terms, we assume that the relevance of each paragraph $g_{xy} \in G$ is defined by a probabilistic distribution over the term set Ω in D^+, which is modelled using the set-valued mapping $\Gamma_1(g_{xy})$. To estimate the term relevance, we assume that the relevance of a term t depends on a probabilistic mixture of G, which is modelled using the inverse set-valued mapping $\Gamma_1^{-1}(t)$. The set G is the evidence space in this case, and a set of terms can represent the relevance of a paragraph g_{xy}, but its relevance level to the entire space is yet unknown as it depends on its local relevance at d_x. Therefore, the probability distribution Ψ_1 is defined on G to indicate this uncertainty; Ψ_1 is then used to estimate the relevance level of g_{xy} to the terms.

Let the probability of a term t relevant to g_{xy} be $P(t|g_{xy})$. Since each paragraph g_{xy} is described by the probability distribution over the set Ω, we have the set-valued mapping Γ_1 to represent and describe the relationship between a set of terms and a paragraph as follows:

$$\Gamma_1 : G \to 2^{\Omega \times [0,1]}$$

such that

$$\Gamma_1(g_{xy}) = \{t \in \Omega | P_{xy}(t|g_{xy}) > \zeta\}$$

where $\Gamma_1(g_{xy}) = \{(t_1, P_{xy}(t_1|g_{xy})), \ldots\}$ for all $g_{xy} \in G$ and ζ is a user-defined threshold assigned to $\zeta = 0$ in this study. Given Ψ_1, as a probability distribution defined on G, we call the pair (Ψ_1, Γ_1) an ERS.

Since there is a need to identify the significance level of a term t, the inverse set-valued mapping of Γ_1 is considered to estimate a representative distribution Ψ_1 on G. For all terms $t \in \Omega$, the inverse set-valued mapping of Γ_1 is defined as

$$\Gamma_1^{-1} : \Omega \to 2^G$$

such that

$$\Gamma_1^{-1}(t) = \{g_{xy} \in G | t \in \Gamma_1(g_{xy})\}$$

to also represent and understand the relationships between a term and a set of paragraphs.

However, while Γ_1^{-1} is used to estimate the significance level of the term t to a subset of paragraphs from G, these paragraphs might not be related to a particular document $d_x \in D^+$, knowing that we assume d_x is the local space used to estimate the relevance of any $g_{xy} \in G$. Therefore, and as in our SIF2 model, $\Psi_1(t)$ is relaxed by considering the relationships between documents and terms. However, the relevance level of d_x to the D^+ is still unknown. Consequently, we define Ψ_2 as a probability distribution over D^+ and propose the set-valued mapping Γ_2 as follows:

$$\Gamma_2 : D^+ \to 2^{\Omega \times [0,1]}$$

to represent each d_x as a probability distribution over all terms in Ω such that

$$\Gamma_2(d_x) = \{t \in \Omega | P_x(t|d_x) > 0\}$$

and the probability of a term t relevant to d_x is $P(t|d_x)$. We also call (Ψ_2, Γ_2) an ERS. Therefore, $\Gamma_2(d_x)$ can be described as

$$\Gamma_2(d_x) = \{(t_1, P_x(t_1|d_x)), \ldots\}$$

To estimate the relevance weight of a term t to the user information needs, which are represented in our study by D^+, the inverse set-valued mapping of Γ_2 is proposed as follows:

$$\Gamma_2^{-1} : \Omega \to 2^{D^+}$$

where

$$\Gamma_2^{-1}(t) = \{d_x \in D^+ | t \in \Gamma_2(d_x)\}$$

We define the scoring function $sr_g(t)$ on Ω such that

$$sr_g :: \Omega \rightarrow \mathbb{R}_{>0}$$

and

$$\mathbb{R}_{>0} = \{sr_g(t) \in \mathbb{R} | sr_g(t) > 0\}$$

as follows:

$$\Psi_1(t) \propto sr_g(t) = \sum_{d_x \in \Gamma^{-1}(t)} \left\{ P_x(t|d_x) \cdot \left[\sum_{g_{xy} \in \Gamma^{-1}(t)} P(g_{xy}) \times P_{xy}(t|g_{xy}) \right] \right\} \tag{4.5}$$

where $P(g_{xy})$ is the probability of g_{xy} being relevant to what the user needs.

As the paragraph g_{xy} can discuss multiple sub-topics or themes, we assume that g_{xy} is a probabilistic mixture of a set of latent topics Z in G, which is modelled using the set-valued mapping $\Gamma_3(g_{xy})$. The topic Z is the evidence space in this case. The set Z can represent the relevance of g_{xy} to the user information needs. The more relevant topics a paragraph covers, the more the paragraph's relevance increases. This implies the relevance of frequent topics (topics shared by many paragraphs). However, the relevance level of g_{xy} to the entire Z space is unknown without estimating the relevance of g_{xy} to the topics in d_x.

Similarly, as before, Ψ_3 is a probability distribution defined on Z to indicate this uncertainty, and Ψ_3 is used to estimate the relevance level of g_{xy} to Z, managed by the pair (Ψ_3, Γ_3). As each paragraph g_{xy} is described by the probability distribution over the set Z of topics, a set-valued mapping of Γ_3 is proposed to represent the relationship between a paragraph and a set of topic as follows:

$$\Gamma_3 : G \rightarrow 2^{Z \times [0,1]}$$

and

$$\Gamma_3(g_{xy}) = \{z_j \in Z | P_{xy}(z_j|g_{xy}) > 0\}$$

where $\Gamma_3(g_{xy}) = \{(z_1, P_{xy}(z_1|g_{xy})), \ldots\}$ for all $g_{xy} \in G$.

However, $P(z_j|g_{xy})$ can only estimate the topical significance of z_j given g_{xy}; we must estimate the relevance of g_{xy} at d_x instead. Therefore, let the probability of a paragraph g_{xy} relevant to a given topic z_j be $P(g_{xy}|z_j)$. Further, Γ_3^{-1} is proposed to describe and measure the strength of the inverse relationship between a topic z_j and the set of paragraphs $S \subseteq G$ as follows:

$$\Gamma_3^{-1} : Z \to 2^G$$

and

$$\Gamma_3^{-1}(g_{xy}) = \{z_j \in Z, g_{xy} \in S | P_j(g_{xy}|z_j) > \xi\}$$

where ξ is another user-defined threshold assigned to $\xi = 0$ in this study. Therefore, the relevance of g_{xy} to d_x can be estimated as follows:

$$\Psi_3(g_{xy}) \propto P(g_{xy}) \propto \sum_{z_j \in \Gamma_3^{-1}(g_{xy})} P_{xy}(g_{xy}|z_j) \tag{4.6}$$

By integrating Equation 4.6 into Equation 4.5, the relevance score of the term t (i.e., $sr_g(t)$) can be calculated as follows:

$$sr_g(t) = \sum_{d_x \in \Gamma_2^{-1}(t)} \left\{ P_x(t|d_x) \times \left[\sum_{g_{xy} \in \Gamma_1^{-1}(t)} \left(\frac{1}{P_{xy}(t|g_{xy})} \cdot \left(\sum_{z_j \in \Gamma_3^{-1}(g_{xy})} P_{xy}(g_{xy}|z_j) \right) \right) \right] \right\} \tag{4.7}$$

To find the latent sub-topics in G, we use LDA, which provides $P(z_j|g_{xy})$. However, we need $P(g_{xy}|z_j)$, which is estimated as follows:

$$P_{xy}(g_{xy}|z_j) = \frac{P(z_j) \times P_{xy}(z_j|g_{xy})}{P_x(z_j|d_x)}$$

Here, $P(z_j|d_x)$ is estimated by the LDA model and $P(z_j)$ is the marginal probability of z_j in G, which can be calculated based on the definition of Γ_3 as follows:

$$P(z_j) = \sum_{z_j \in \Gamma_3(g_{xy})} P_{xy}(z_j|g_{xy})$$

4.2.3.2 Global Relevance

The scoring function $sr_g(t)$ can be used to estimate the local relevance of a paragraph $g_{xy} \in G$, using the relevance of its terms to the user information needs. Therefore, as indicated before, the global relevance of g_{xy} to the complete user information needs that are discussed across D^+ documents can then be calculated through the summation of its local relevance in every $d_x \in D^+$ as follows:

$$S_G(t) = \sum_{t \in d_x, d_x \in D^+} \Bigg(\sum_{g_i \in d_x, t \in g_i} sr_{g_i}(t) \Bigg) \tag{4.8}$$

where $sr_{g_i}(t)$ estimates the relevance of term t of paragraph g_i in document $d_x \in D^+$.

However, while $S_G(t)$ can estimate the global relevance of the paragraphs in D^+, using this relevance to reduce uncertainties in relevant features that are discovered by various TFS models and frameworks without losing the qualities of the originally discovered features must be addressed. Therefore, in the next section, we address the issue of adopting the proposed UR method using a two-step tactic, by 1) scaling the relevance of a selected feature (e.g., a weighted term t) and 2) re-ranking the scaled set of relevant features.

4.2.4 Re-Ranking Relevant Features

To effectively represent user information needs, we first must select a set of terms that are representative. To find such terms, a TFS model is selected, such as SVM [Dumais et al., 1998]. As a discriminative classifier, SVM finds a hyperplane that best separates the positive and the negative classes. The discrepancy between normal values and the hyperplane is used to weight and thus rank the terms, and then a subset is empirically selected from these ranked terms. Since SVM and other existing models consider a given document relevant if some parts of the document are relevant, some terms selected by these models can come from irrelevant or less relevant parts of the document. Therefore, the selected terms, their weights and their ranks incorporate uncertainties. We aim to reduce these uncertainties by effectively scaling the term weights and re-ranking the terms based on their relevance value estimated by Equation 4.8.

Let the weight of a term t estimated by a model (e.g., SVM) be $w_m(t)$ and its relevance, estimated by Equation 4.8, be $S_G(t)$. The re-ranking weight (i.e., score) $w(t)$ of the term is estimated by scaling $w_m(t)$ by $S_G(t)$ as follows:

$$w(t) = w_m(t) \times S_G(t) \tag{4.9}$$

Then, the terms are re-ranked based on the new weight $w(t)$. When re-ranking is applied to the model (e.g., SVM), we call it the improved iModel (e.g., iSVM). An intuitive interpretation of $w(t)$ is that it combines the paragraph-level relevance evidence with the document-level relevance evidence, which is estimated by the existing models for reducing uncertainty. However, the sentence-level evidence is too specific, and our preliminary experiments showed that such evidence is not effective in our current relevant term re-ranking model.

4.3 Chapter Summary

This chapter presented SIF2, an innovative fusion-based model for selecting informative topical terms from a collection of documents that discusses user information needs. The model extends multiple random sets to fuse hierarchical LDA-based features and accurately weight topical terms on a document-by-document basis. SIF2 also combines the aggregated topical terms' weights with their document frequencies to estimate a global score. This fused global score more accurately reflects the informativeness of a term to the key topics of interest discussed in the collection. The experimental results (see Section 6.8.2) demonstrated that SIF2 attained significant performance improvements in IF and RRT experiments compared to all baseline models. SIF2 demonstrates an effective hybrid fusion strategy for integrating the advantages of unsupervised topic modelling and collection statistics.

This chapter also addressed the challenge of reducing uncertainties in relevant feature space by using implicit paragraph relevance. The proposed UR method uses topics in relevance feedback discovered by LDA to estimate the implicit paragraph relevance. Multiple ERSs are used to model the complex relationships between features, paragraphs and topics, and to deal with the associated uncertainties. The experimental results (see Section 6.8.3) confirm the proposed UR method's merit as a feature re-ranking technique for relevance discovery. The substantial improvement achieved by applying the proposed method is due to the effective estimation of paragraph relevance, as well as its use in estimating feature relevance. This research's theoretical contribution regards using multiple ERSs for modelling uncertainties associated with the complex relationships between features, paragraphs and topics as essential entities in the feature weight-scaling process. This study provides a promising methodology for combining paragraph-level evidence with document-level evidence to estimate feature relevance.

Despite the effectiveness of the proposed SIF, SIF2 and UR models, they are biased towards the most frequent topics or themes in the collection. Highly frequent topics can overshadow less frequent but equally important ones, which makes it challenging to identify relevant features that precisely describe the user information preferences. Moreover, the three fusion models also cannot deal with relevant features that frequently appear in both positive and negative feedback documents. Therefore, in the next chapter, two frameworks will be introduced to deal with the limitations mentioned above by treating feature selection and feature weighting as two independent tasks. To do this, the proposed frameworks will integrate different supervised and unsupervised learning algorithms in addition to our SIF and UR models to select and then re-weight relevant features that describe user information needs.

Chapter 5

Hybrid Fusions Frameworks for Relevant Feature Discovery

This chapter describes two innovative and highly-effective frameworks that were developed to identify relevant topical terms [1] that reflect user information needs. The frameworks integrate different learning algorithms and multiple hybrid fusion-based modules, which were developed based on our SIF and UR models, to select and then re-weight topical terms at two separate stages of features fusion. The first unsupervised framework is known as USIF. This framework was especially developed to address LDA bias towards frequent topics in a collection of documents that can undermine less frequent but equally important topics. The second supervised framework is known as SSIF. The SSIF framework was developed to manage the effects of topical terms that appear repeatedly in both positive and negative user relevance feedback. Section 5.2 and Section 5.1, describe the USIF framework and SSIF framework, respectively. Section 5.3 provides a summary of this chapter. Both frameworks are evaluated in relation to their IF and RRT applications. The experimental results are presented in Chapter 6.

5.1 The Proposed USIF Framework

5.1.1 Introduction

As described above, relevant feature discovery aims to identify a set of representative features (feature selection) and estimate their relevance (feature weighting) in relation to a user's topics of interest in a collection of relevant documents [Gao et al., 2015, Li et al., 2015, 2010, 2012]. As noted above, as topic modelling algorithms [Blei, 2012, Blei et al., 2003, Hofmann, 2001]

[1] The terms ’topical terms’, ’lexical features’, ’term’ and ’features’ are used interchangeably in this chapter.

are the only algorithms that explicitly assume that documents can exhibit multiple topics, they are most suited to discovering relevant features [Blei et al., 2003, Gao et al., 2014b, 2015]. Whether supervised or unsupervised, most relevance discovery techniques, including topic-based models, conduct the selection and weighting of relevant features as dependent tasks [Manning et al., 2008b, Robertson and Zaragoza, 2009, Yang and Pedersen, 1997]. However, adopting a data fusion perspective, this study argued that treating feature selection and weighting dependently may be ineffective given the uncertainties in training collections and that most these collections are topically unbalanced [Alharbi et al., 2018a, Lewis et al., 2004]. Notably, the use of sequential closed pattern mining to select some representative features (i.e., patterns) has been effective in reducing noisy and redundant features [Li et al., 2015, 2010, Wu et al., 2006]. However, the adoption of interestingness measures (i.e., support and confidence) in pattern mining algorithms to estimate the relevance of these representative patterns has considerably undermined their effectiveness in representing user information needs and led to undesirable results [Li et al., 2015, 2011].

Both supervised and unsupervised relevance discovery algorithms are affected by uncertainties in the relevant documents [Alharbi et al., 2018a]. Notably, supervised algorithms require large sets of manually-labelled training documents that may be labour expensive and time consuming [Algarni, 2011, Soleimani and Miller, 2016]. Conversely, unsupervised algorithms, particularly probabilistic topic modelling algorithms, are biased towards the most dominant topics in a document collection (i.e., topics that are shared by many documents in the collection). However, even topics that are only briefly discussed in documents could be important to users' needs [Alharbi et al., 2017b, Anastasiu et al., 2013, Jain, 2010]. Additionally, these methods also appear to favour frequent sub-topics (i.e., themes) of a particular general topic of interest; however, this can make it challenging to capture the thematic relevance of the features if these themes are randomly discussed at the paragraph level [Alharbi et al., 2018a, Chien, 2016]. Thus, under an unsupervised framework, it is challenging to select representative features, as frequent topics or themes may overshadow less frequent but equally relevant themes. Additionally, can also be challenging to accurately weight these features, as they may be unevenly distributed across the relevant documents and paragraphs in a collection.

The unsupervised technique of clustering has widely been used to gain an understanding of unlabelled data and to facilitate the discovering of knowledge from document collections [Anastasiu et al., 2013, Jain, 2010]. Document-clustering algorithms group similar documents into clusters according to specified similarity measures [Aggarwal and Zhai, 2012, Anastasiu et al., 2013]. For many years, document clustering has been used in retrieval systems to organise documents around a single subject or topic. Such cluster-based language models represent a significant improvement over standard document-based models [Kozorovitsky and Kurland, 2011a, Krikon and Kurland, 2011, Liu and Croft, 2004]. However, the assumption that a cluster of documents describes only one topic may be too simple given that most long documents discuss multiple topics and themes. As the document-clustering algorithm does not depend on the frequency of topics in documents to form a cluster of similar documents [Aggarwal and Zhai, 2012, Li et al., 2016], it can be used to limit the impact of frequent relevant topics by treating each cluster in the collection as equally important. However, unlike topic models, clustering does not provide details of the topics in each cluster, as these topics are hidden in the clusters of the documents. Additionally, the clustering algorithm does not explicitly provide a way to select or weight the relevant features that may appear in a cluster (i.e., intra-cluster features) or across all clusters (i.e., inter-cluster features). This study sought to address the following question: Is there a method that effectively incorporates the advantages of document clustering and topic modelling to discover the relevant features that effectively represent user information needs?

In this section, we present our innovative USIF framework [2]. This framework integrates document clustering and topic modelling to select and then re-weight relevant topical terms that describe users' information preferences at two *independent* stages. As Figure 5.1 shows, the USIF framework uses multiple fusion modules that were developed based on the theoretical foundations of our SIF and UR models. In the first stage, the USIF framework selects a ranked set of representative, inter-cluster, topical terms using an elegant method that conceptually agglomerates relevant clusters in a taxonomic style and selects the features at a specific level of abstraction. This step ensures that the selected features are not biased towards frequent topics, as each cluster is considered equally important. The conceptual agglomeration algorithm is also integrated with our ERS-based SIF model to uncover each cluster's hidden topics and estimate the topical relevance of the intra-cluster features before the selection process occurs. In the second stage, the framework estimates the relevance of the selected topical terms based on the fusion of their topical and thematic significances and their global representativeness across all

[2] Parts of this framework were published in [Alharbi et al., 2017b] and [Alharbi et al., 2018a]. The abbreviation 'USIF' stands for **U**nsupervised **S**election of **I**nformative **F**eatures.

of the documents in the collection. Finally, the framework uses the fused score estimated in the second stage to re-weight the selected, ranked topical terms identified in the first stage. The results of experiments, which were conducted using the first 50 collections of documents from the standard RCV1 dataset and TREC filtering topics, show that our USIF framework is highly effective. It significantly outperforms state-of-the-art supervised and unsupervised models as presented in Section 6.8.4 and analysed and discussed in Section 6.9.4.

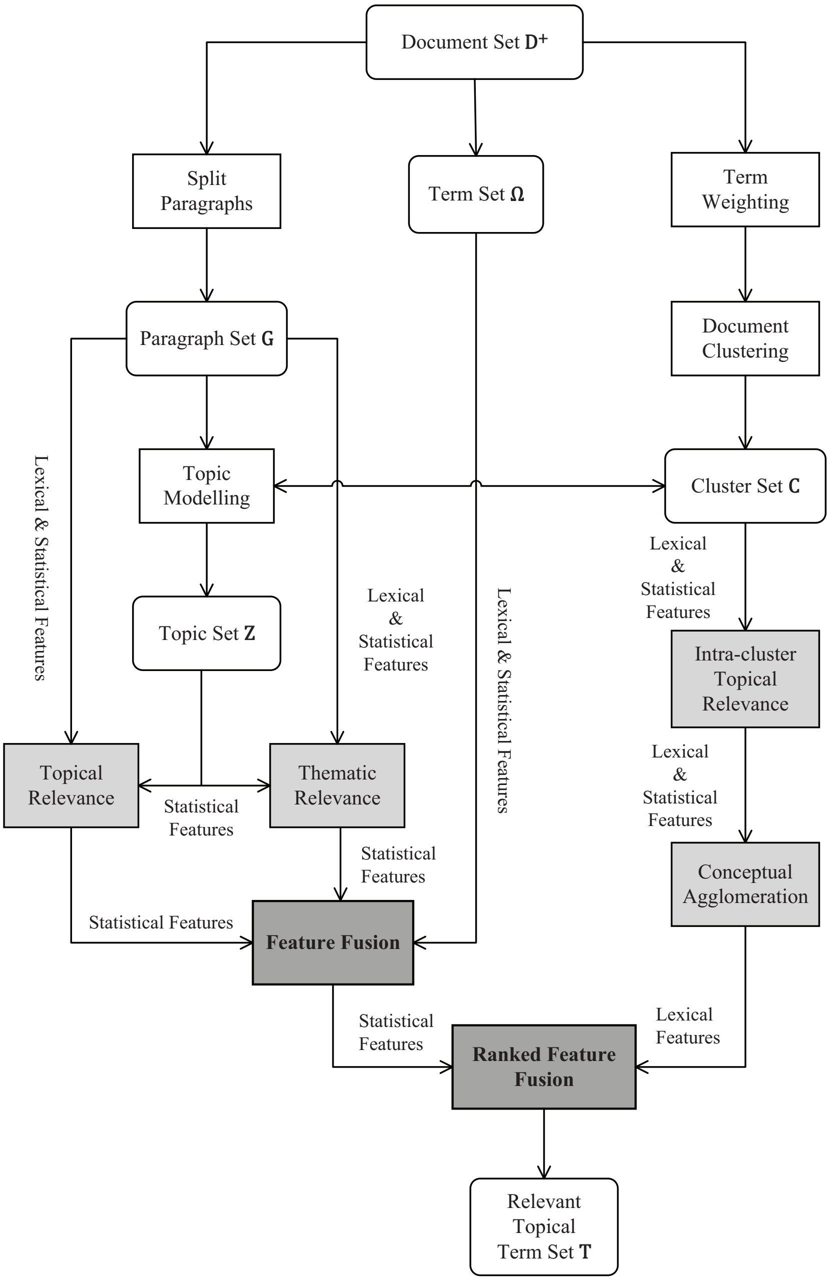

Document Set $\mathbf{D}^+$
Split Paragraphs
Term Set $\boldsymbol{\Omega}$
Term Weighting
Paragraph Set $\mathbf{G}$
Document Clustering
Topic Modelling
Cluster Set $\mathbf{C}$
Lexical & Statistical Features
Topic Set $\mathbf{Z}$
Lexical & Statistical Features
Lexical & Statistical Features
Lexical & Statistical Features
Intra-cluster Topical Relevance
Topical Relevance
Statistical Features
Thematic Relevance
Lexical & Statistical Features
Statistical Features
Conceptual Agglomeration
Statistical Features
Feature Fusion
Statistical Features
Ranked Feature Fusion
Lexical Features
Relevant Topical Term Set $\mathbf{T}$

Figure 5.1: The structure of the USIF framework.

Figure 5.1 not only illustrates the fusion modules of our USIF framework, but also depicts the used entities; that is, the relevant document clusters C, the set of paragraphs G in the collection D^+, their topics Z and vocabulary list Ω. This figure also shows the flow of the adopted lexical (terms) and statistical features by the framework's fusion modules. Additional details about the proposed USIF framework are described in the following sections. First, Section 5.1.2 discusses the problem formulation. Next, Section 5.1.3 provides an overview of unsupervised learning algorithms. Following this, Section 5.1.4.1 describes the framework's first stage and Section 5.1.4.2 outlines the details of the second stage of the USIF algorithm. Next, Section 5.1.5 describes the fusion of the framework's two stages of the USIF algorithm. Finally, Section 5.1.6 outlines the time complexity analysis.

5.1.2 Problem Formulation

It was assumed that a user has a collection of long documents D^+ that are pertinent to the subject of *economic espionage* and its related topics of interest, such as *industry espionage*, *technical espionage*, *commercial espionage* and *corporate espionage*. To further investigate this subject, the user wishes to enrich the collection by gathering more relevant documents from the Web. To achieve this goal, the researcher needs a relevant feature discovery framework that can accurately select and give weight to a representative set of topical terms that effectively describe the collection. The user can then use these weighted terms to collect the required relevant documents. However, it should be noted that such topics of interest are not generally evenly distributed in a collection in which some topics are frequent and other topics are non-frequent.

Frequent topics refer to topics featured in many documents in a collection. Conversely, non-frequent topics refer to topics featured in a lower number of documents. Many equally important topics may be non-frequent, as a collection may not have sufficient documents to determine the optimal frequency of these topics. LDA is an effective tool for discovering latent topics in a corpus that are different to those topics of interest (see above). However, LDA favours the most frequent topics; for example, the generated topics might be more relevant to the topic of interest *commercial espionage*, as it is featured in most documents in the collection. Consequently, many useful but non-frequent topics are overshadowed by frequent topics; however, this makes both the selection and weighting of the features described by these less frequent topics rather challenging. This problem is further complicated in relation to long documents, as it is highly likely that a topic may have multiple and unbalanced themes (i.e., sub-topics). Further, as the long documents may suffer from uncertainties related to irrelevant or not very relevant paragraphs (see Figure 1.3) and other LDA-related problems (see discussions in previous chapters), the necessity of a more holistic solution, which addresses all these problems in the form of a framework for relevant feature discovery, increases.

One possible solution to the problem of the LDA bias towards the frequent topics of interest is to group the documents of the collection into clusters based on their similarities. Each cluster identifies a topic regardless of the frequency of the documents that discuss this topic in the collection. Each cluster is treated equally to limit the effect of frequent topics. Next, the clusters in the collection are conceptually agglomerated in a taxonomic style to select a set of topical terms that are representative of all topics in the clusters. However, the selected terms might not reflect the detailed topics and themes in the collection, as most traditional clustering algorithms assume that a cluster describes a single topic; however, this approach may be ineffective, as long documents often discuss multiple topics and themes. Thus, LDA was adapted and used to discover the hidden topics and themes in every cluster and estimate the informativeness of each topical term based on its topical and thematic significances in the original collection rather than on any artificially formed clusters. The purpose of using document clusters to select representative topical terms is to reduce the bias of topic modelling towards frequent topics. In the following section, a brief description is provided of document-clustering and the LDA model in relation to two well-known unsupervised learning algorithms.

5.1.3 Background Overview

In the first stage of the proposed USIF framework, the relevant document set D^+ is statically organised into groups (aka clusters) using a clustering algorithm that is based on similarity (aka distance) measures [Huang, 2008]. This study assumes that a relevant long document d has a set of paragraphs and that each paragraph contains a bag of terms. The set G is the set of all paragraphs in D^+. Additionally, a cluster c_r in this study is considered a subset of relevant documents that share a similar topic of interest. Thus, $cluster(D^+) = \{C_1, C_2, \ldots, C_r\}$, such that $C_r = \{d_x : x \leq M, d_x \in D^+\}$, where $M = |D^+|$, L is the total number of clusters in D^+ that is automatically identified by a document-clustering algorithm and thus $C_r \subseteq D^+$.

5.1.3.1 Document Clustering

Clustering D^+ was completed in the first stage of our framework using the bisecting K-means (BKM) algorithm [Steinbach et al., 2000], which uses a partitional clustering technique. This algorithm is widely used by researchers to cluster large document collections because of its low computational overheads [Anastasiu et al., 2013, Beil et al., 2002, Savaresi and Boley, 2001]. The BKM algorithm groups similar documents together in a cluster by maximising the intra-cluster similarity between documents and minimising the similarity between each inter-cluster (i.e., by maximising the inter-cluster distance). The documents in our framework are represented in the vector space model as BoW. The BKM algorithm requires that pairwise document similarity be calculated using some distance measures, such as the Euclidean distance, cosine similarity, the Jaccard coefficient and the Pearson correlation coefficient [Steinbach et al., 2000]. Our USIF framework uses cosine similarity as the distance measure used by the BKM algorithm, as it is the most widely used similarity measure and has been shown to work effectively with the BKM algorithm [Steinbach et al., 2000]. The BKM algorithm also requires that the number of clusters L be specified beforehand. However, it is challenging to specify the optimal number of clusters accurately [Das et al., 2008, Jain, 2010]. In our model, we do not assume that the number of clusters would be optimal; rather, a trial-error approach is adopted in our experiment. Section 6.7 provides further details about how we experimentally predetermined the number of clusters for a collection of documents.

5.1.3.2 Topic Modelling

In both stages and for both the D^+ collection and each cluster $C_r \subseteq D^+$, our USIF framework uses LDA to reduce the dimensionality of the relevant documents' paragraphs in G to a set of manageable topics Z where V is the number of topics. In accordance with [Gao et al., 2015], each paragraph $g_y \in G$ is assumed to contain multiple latent topics. As mentioned above, LDA defines each topic $z_j \in Z$ as a multinomial probability distribution over all terms in D^+ or C_r as $p(t_i|z_j)$ in which Ω represents all terms in D^+, $t_i \in \Omega$ and $1 \leq j \leq V$, such that $\sum_i^{|\Omega|} p(t_i|z_j) = 1$. LDA also represents each individual paragraph in G as a probabilistic mixture of topics as $p(z_j|g)$. As a result, and based on the number of latent topics, the probability (local weight) of term t_i in paragraph g_y is calculated by $p(t_i|g_y) = \sum_{j=1}^{V} \left(p(t_i|z_j) \times p(z_j|g_y)\right)$. Finally, all hidden variables, $p(t_i|z_j)$ and $p(z_j|g)$, are statistically estimated by the Gibbs sampling algorithm [Steyvers and Griffiths, 2007].

In the current literature (e.g., [Bashar and Li, 2017, Bashar et al., 2016, Gao et al., 2014b, 2015]), each topic z_j is represented with the top-k terms sorted in descending order by $p(t_i|z_j)$. These top-k terms in z_j are closely related to topic z_j and there are V such topics. This kind of representation is effective in the analysis of individual topics; however, this kind of topic representation is not effective in estimating the topical relevance of features for representing user information needs. If terms in a topic are discarded that are not in the top-k list, important information may be missed. Thus, instead of representing each topic by its top-k features, we use multiple ERS to model the complicated and imprecise relationship between the terms, topics and the relevant collections' paragraphs and to estimate the topical and thematic relevance of the collection's topical terms.

5.1.4 USIF Fusion Stages

Unlike traditional unsupervised relevance discovery models, the proposed USIF framework differentiates between the selection and weighting processes of relevant features by using two independent feature fusion stages. In the absence of a search guide and labelled training set and given the existence of uncertainties, this differentiation approach facilitates the effective fusion of different lexical and statistical features that are independently discovered and estimated at each stage. Thus, the selection task focuses on specific aspects, such as identifying representative topical terms from a set of equally relevant clusters, while the weighting task accurately estimates a more accurate fused score for each of these topical terms using entities in the collection other than the artificially formed clusters. The following two sections provide further details about each stage of the proposed USIF framework.

5.1.4.1 Stage 1: Topical Term Selection

As noted above, a document-clustering algorithm is used in the first stage of our USIF framework to alleviate the impact of frequent topics in the document collection and thus limit the bias of LDA towards these topics. The formed clusters are then used as leaf nodes in a hierarchical taxonomy that is conceptually agglomerated during the topical term selection. Several studies [Blei et al., 2010b, Chien, 2016, Weninger et al., 2012] have used taxonomy models to represent topics and documents of a corpus. A hierarchical taxonomy is a common technique whereby items are conceptually grouped into increasingly smaller granularities within which each non-leaf node is a conceptual agglomeration of its siblings [Cai and Hofmann, 2004, Weninger et al., 2012]. A node in a taxonomy can be described as the sum of its super-node features and node-specific modifier features [Hwang and Sigal, 2014]. This implies that the features found on the path from the root to the leaf describe the leaf (the cluster) [Petinot et al., 2011]. The biological classification of species is a good example of a taxonomy under which species are placed only at the leaf nodes, while the inner nodes, such as those for primates and mammals, conceptually agglomerate the species. The path of each species through the taxonomy can be used to describe such species.

Inter-Cluster Topical Term Selection

Figure 5.2 shows the structure of our taxonomic selection model, where c_r is a cluster, a_n is a non-leaf node that conceptually agglomerates clusters and t_i is a topical term. In this taxonomy, for example, a_4 is the conceptual agglomeration of the clusters c_1, c_2, a_2 is the conceptual agglomeration of a_4 and a_5 and a_1 is the conceptual agglomeration of a_2 and a_3. The node a_1 at abstraction level three is described by the topical term t_1, and a_1 conceptually agglomerates all the clusters (c_1 to c_8). This means that all the clusters share this topical term t_1. The node a_2 at abstraction level two is described by its node-specific topical terms t_2 and t_3 and the super-node topical term t_1 and a_2 conceptually agglomerates the clusters from c_1 to t_4. This means that the topical terms $\{t_1, t_2, t_3\}$ are shared by the clusters from c_1 to c_4. Thus, the nodes in higher abstraction levels are more general and have fewer topical terms, while the nodes in lower abstraction levels are more specific and have more topical terms. The abstraction level is determined based on the application, the topical terms required to describe the nodes at that level are then selected as the representative topical terms of the given collection.

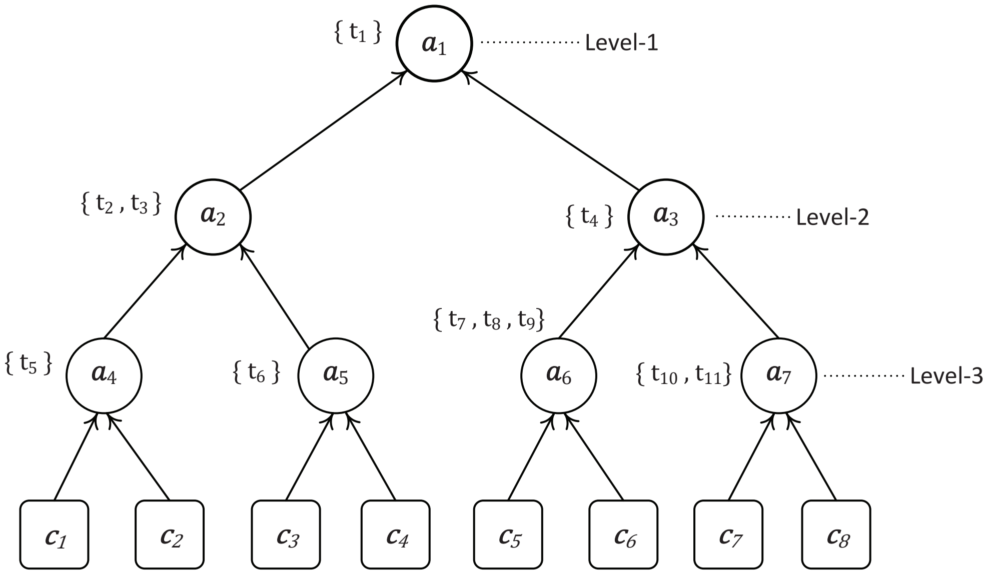


Figure 5.2: The conceptual agglomeration of relevant clusters.

Intra-Cluster Topical Term Selection

As mentioned above, the assumption that a cluster of long documents can only discuss one topic is rather simple, as a sample document may include multiple related sub-topics or themes (see Figure 1.1). Thus, these hidden sub-topics need to be uncovered and the topical relevance of any terms that appear frequently across the cluster's documents (i.e., the intra-cluster topical terms) need to be estimated. For this task, our SIF model was applied to each cluster. Some systems may ask for the top-k representative topical terms rather than terms at the abstraction level. For example, an IF system may ask for the top six terms from Figure 5.2. If we select level two, only four terms ($\{t_1, t_2, t_3, t_4\}$) are identified. Conversely, if we select level one, the 11 terms depicted in the figure are identified; however, such a high number is more than required. In this case, we select all the topical terms required to describe the nodes in level two (the lowest *full-level*) and are given four terms. The remaining terms from level one (the highest *partial-level*) are then selected using the score fusion function $r(t_i)$ (as described in the following section). The following section also discusses the second stage of our USIF framework in which the topical and thematic relevance of the inter-cluster topical terms are estimated based on their appearance in the entire relevant documents of the collection.

5.1.4.2 Stage 2: Topical Term Weighting

In the first stage of the USIF framework, a set of representative, inter-cluster topical terms are selected via the integration of document-clustering and topic modelling (as determined by the proposed conceptual agglomeration algorithm). Our previously proposed SIF model was used to relax the single topic assumption of the clustering approach and select the most representative intra-cluster topical terms. The conceptual agglomeration of equally relevant clusters was used to effectively select those terms that represent the essential topics discussed across all the formed clusters. However, the estimated topical relevance of each term in each cluster could not be generalised due to the unbalanced set of clusters. Thus, in the second stage, the thematic and topical significances of each inter-cluster term are re-estimated based on its original appearance in the collection. To do this, the theoretical merits of our SIF model and the UR method are used.

Term Thematic Significance

Themes refer to the main ideas of a document set and are implicitly expressed across paragraphs [Chien, 2016]. Thus, paragraphs are used to capture the thematic relevance of terms. Thematic relevance captures the general focus of user information needs. Let G be the set of paragraphs in the relevant documents D^+. Each paragraph $g_y \in G$ is a probabilistic distribution over the term space Ω, which is modelled using set-valued mapping $\Gamma_1(g_y)$. It is assumed that a term's t_i thematic relevance is a probabilistic mixture of G, which is modelled using the inverse set-valued mapping $\Gamma_1^{-1}(t_i)$. Figure 5.3 shows all the proposed set-valued mappings.

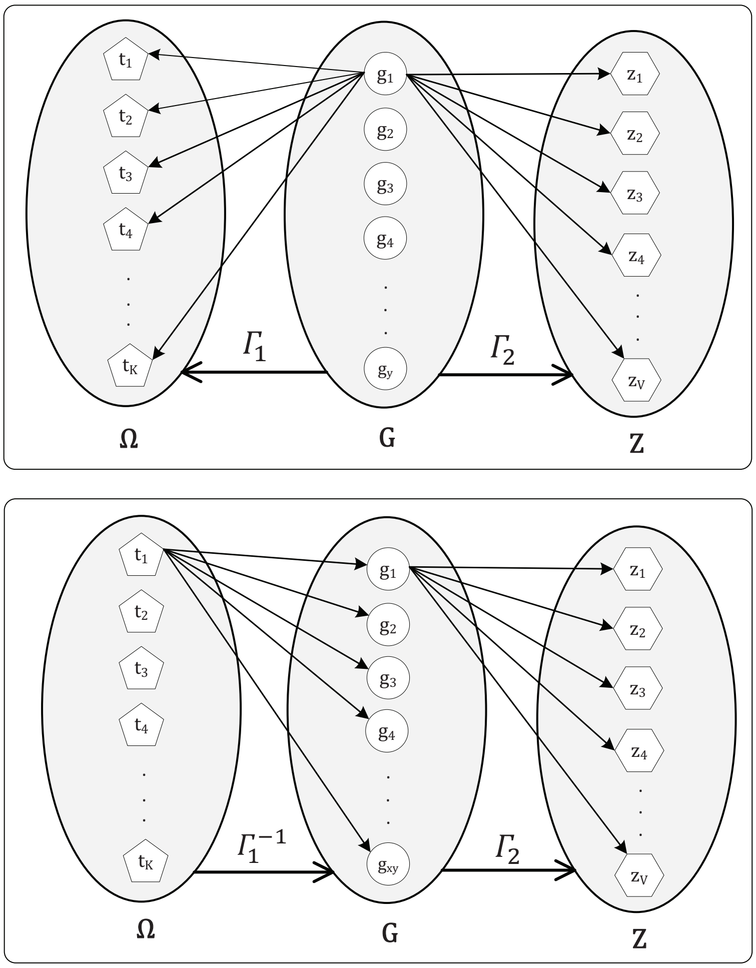


Figure 5.3: The mappings of Γ and Γ^{-1} for estimating the thematic significance of terms.

The set G is the evidence space and a set of terms represents a paragraph g_y; however, a term's relevance level to the evidence space is unknown. Thus, the probability distribution Ψ_1 is defined using G to indicate this uncertainty. Let the probability of a term t_i be relevant to g_y be $P(t_i|g_y)$, where, for simplicity, it is assumed that $P(t_i|g_y) = 1$ if $t_i \in g_y$ and $P(t_i|g_y) = 0$ if $t_i \notin g_y$. Next, ERS (Ψ_1, Γ_1) is used to model and describe the relationship between the paragraphs and terms. As each paragraph g_y is described by the probability distribution over the set Ω, the set-valued mapping of

$$\Gamma_1 : G \to 2^{\Omega\times[0,1]} - \{\emptyset\}$$

such that

$$\Gamma_1(g_y) = \{t_i \in \Omega | P_y(t_i|g_y) > \zeta\}$$

is proposed to represent and describe the relationship between a set of terms and a paragraph, where $\Gamma_1(g_y) = \{(t_1, P_y(t_1|g_y)), \ldots\}$ for all $g_y \in G$ and ζ is a user-defined threshold assigned to $\zeta = 0$.

As there is a need to identify the relevance level of a selected term t_i, the inverse set-valued mapping of Γ_1 is considered to estimate a representative distribution Ψ_1 on G. For all terms $t_i \in \Omega$, the inverse set-valued mapping of Γ_1 is defined as

$$\Gamma_1^{-1} : \Omega \to 2^G$$

such that

$$\Gamma_1^{-1}(t_i) = \{g_y \in G | t_i \in \Gamma_1(g_y)\}$$

to represent and understand the relationships between a term and a set of paragraphs. Thus, the thematic relevance weight $w_g(t_i)$ of a term t_i to a user's information needs can be estimated as follows:

$$\Psi_1(t_i) \propto w_g(t_i) \propto \sum_{g_y \in \Gamma_1^{-1}(t_i)} P_y(t_i|g_y) \times P(g_y) \tag{5.1}$$

where $P(g_y)$ is the probability of g_y being significantly relevant to the main themes that describe what the user wants (see discussion below).

As paragraph g_y may discuss multiple sub-topics (i.e., themes), it is assumed that g_y is a probabilistic mixture of a set of latent topics Z in D^+, which is modelled using set-valued mapping $\Gamma_2(g_y)$. In this case, Z is the evidence space. The set Z represents the relevance of g_y to the user's information needs. The more relevant topics a paragraph covers, the more the paragraph's relevance increases. This motivation implies the relevance of frequent topics (i.e., topics shared by many paragraphs). However, the relevance level of g_y is unknown. Similarly, as before, Ψ_2 is a probability distribution defined on Z to indicate this uncertainty. The pair (Ψ_2, Γ_2) represents an ERS that models the complex relationship between paragraphs and latent topics.

Let the probability of a paragraph g_y be relevant to a given topic z_j be $P(g_y|z_j)$. As each paragraph g_y is described by the probability distribution over the set Z of topics, there is a set-valued mapping of

$$\Gamma_2 : G \to 2^{Z \times [0,1]} - \{\emptyset\}$$

such that

$$\Gamma_2(g_y) = \{z_j \in Z | P_y(g_y|z_j) > \xi\}$$

where $\Gamma_2(g_y) = \{(z_1, P_y(g_y|z_1)), \ldots\}$ for all $g_y \in G$ and ξ is another user-defined threshold assigned to $\xi = 0$ in this study. Thus, the relevance of g_y to D^+ is estimated as follows:

$$\Psi_2(g_y) \propto P(g_y) \propto \sum_{z_j \in \Gamma_2(g_y)} P_y(g_y|z_j) \tag{5.2}$$

Using Equation 5.1 and Equation 5.2, the thematic relevance weight $w_g(t_i)$ of the term t_i is calculated as follows:

$$w_g(t_i) = \sum_{g_y \in \Gamma_1^{-1}(t_i)} \left\{ P_y(t_i|g_y) \times \sum_{z_j \in \Gamma_2(g_y)} P_y(g_y|z_j) \right\} \tag{5.3}$$

To identify the latent topics in D^+, the LDA was used to estimate $p(z_j|g_y)$; however, $P(g_y|z_j)$ is needed. By applying Bayes' theorem, it is found that $P_j(g_y|z_j) = \frac{p(z_j|g_y) \times p(g_y)}{p(z_j)}$. In this instance, $p(g_y)$ is a prior distribution that can be ignored and $p(z_j)$ is the marginal probability of z_j in G.

Term Topical Significance

Topics are specific matters in a general subject in a collection. Topical relevance captures the specific focus of user information needs. In accordance with topic modelling, it is assumed that each topic z_j is defined by a probabilistic distribution over the terms in the vocabulary Ω, which is modelled with the set-valued mapping $\Gamma_3(z_j)$. In estimating the topical relevance, it is assumed that the topical relevance of a term t_i comes from a probabilistic mixture of a set of topics Z, which is modelled with the inverse set-valued mapping $\Gamma_3^{-1}(t_i)$. Figure 5.4 shows the proposed set-valued mappings.

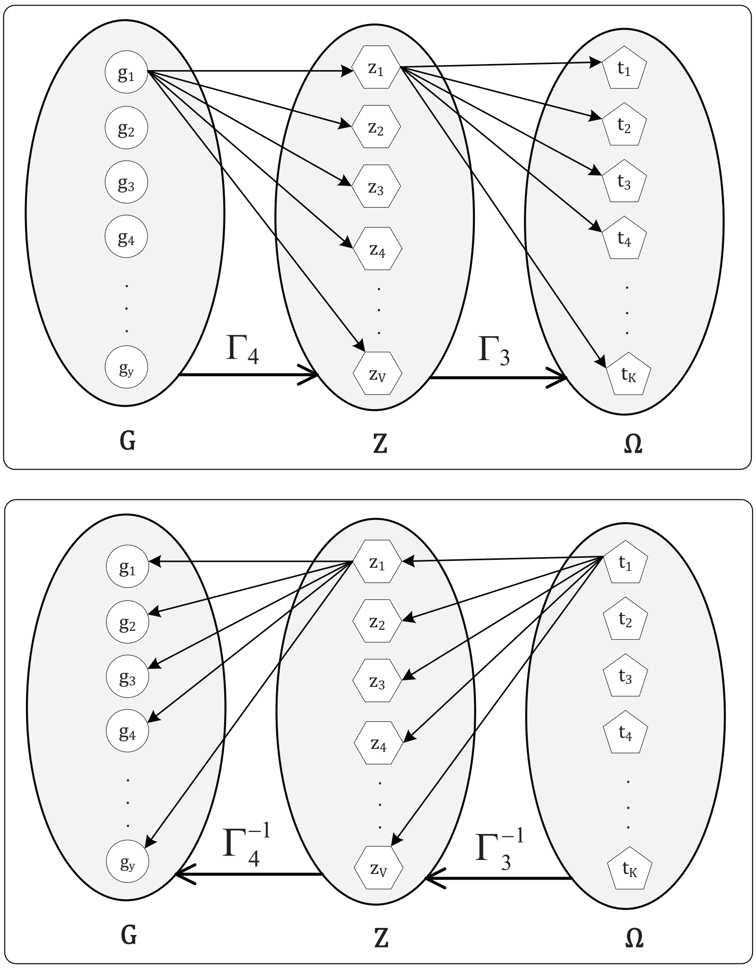


Figure 5.4: The mappings of Γ and Γ^{-1} for estimating the topical significance of terms.

Additionally, similar to topic modelling, it is assumed that a paragraph g_y is a probabilistic mixture of a set of topics Z, which is modelled with the set-valued mapping $\Gamma_4(g_y)$. It is also assumed that frequent topics (i.e., topics featured in many paragraphs) are important as, they are more likely to discuss the general subject in the collection. The relevance of a topic is defined by a probabilistic mixture of a set of paragraphs G, which is modelled with inverse set-valued mapping $\Gamma_4^{-1}(z_j)$.

In this case, the set Z is our evidence space. A set of topics can represent the topical relevance of the selected term t_i, but the relevance level remains unknown. Thus, Ψ_3 is defined as a probability distribution on the specified evidence space to represent this uncertainty. Ψ_3 is also used to find the relevance level of the term. As there is Ψ_3, as probability distribution defined on the evidence space Z, then the pair (Ψ_3, Γ_3) is an ERS. As each topic z_j is described by the probability distribution over the set of terms Ω, there is a set-valued mapping of

$$\Gamma_3 : Z \to 2^{\Omega \times [0,1]} - \{\emptyset\}$$

$$\Gamma_3(z_j) = \{t_i \in \Omega | P_j(t_i|z_j) > \varsigma\}$$

where $\Gamma_3(z_j) = \{(t_1, P_j(t_1|z_j)), \ldots\}$ for all $z_j \in Z$ and ς is assigned as ‘0’ in this research.

We also need to determine the relevance level of the term t_i. Thus, we had to consider the inverse set-valued mapping of Γ_3 to estimate a suitable distribution for Ψ_3 on Z. For all terms $t_i \in \Omega$, the inverse set-valued mapping of Γ_3 is defined as

$$\Gamma_3^{-1} : \Omega \to 2^Z$$

$$\Gamma_3^{-1}(t_i) = \{z_j \in Z | t_i \in \Gamma_3(z_j)\}$$

Thus, the topical relevance weight $w_z(t_i)$ of the term t_i is estimated as follows:

$$\Psi_3(t_i) \propto w_z(t_i) \propto \sum_{z_j \in \Gamma_3^{-1}(t_i)} \big(P_j(t_i|z_j) \times P(z_j)\big) \tag{5.4}$$

where $P(z_j)$ is the marginal probability distribution of z_j over paragraph set G. If $P(t_i|z_j)$ is normalised, then $\Psi_3(t_i)$. This is the same as the marginal probability distribution $P(t_i)$ over the evidence space.

However, the distribution $P(z_j)$ is unknown. To estimate $P(z_j)$, our next evidence space G is considered. A set of paragraphs can define the relevance of a topic z_j in the collection; however, as before, the relevance level remains unknown. Thus, Ψ_4 is defined as a probability distribution on the evidence space G to indicate this uncertainty. Thus, the pair (Ψ_4, Γ_4) is an ERS and is defined on the evidence space G. As each paragraph g_y is defined as a mixture of topics Z in the collection, the set-valued mapping of Γ_4 is defined as as

$$\Gamma_4 : G \to 2^{Z \times [0,1]} - \emptyset$$

such that

$$\Gamma_4(g_y) = \{z_j \in Z | P_y(z_j|g_y) > 0\}$$

where $\Gamma_4(g_y) = \{(z_1, P_y(z_1|g_y)), \ldots\}$ for all $g_y \in G$.

As the relevance level of a topic z_j needs to be determined, the inverse set-valued mapping of Γ_4 must be considered to obtain a probability distribution that suits Ψ_4 on G. For all topics $z_j \in Z$, the inverse set-valued mapping of Γ_4 is defined as

$$\Gamma_4^{-1} : Z \to 2^G$$

such that

$$\Gamma_4^{-1}(z_j) = \{g_y \in G | z_j \in \Gamma_4(g_y)\}$$

The probability distribution Ψ_4 is proportional to the relevance of a topic that is estimated as follows:

$$\Psi_4(z_j) = P(z_j) \propto \sum_{g_y \in \Gamma_4^{-1}(z_j)} \big(P_y(z_j|g_y) \times P(g_y)\big) \tag{5.5}$$

where $P(g_y)$ is the probability distribution of g_y over the given collection. In this research, it is assumed that $P(g_y)$ is equally likely for all $g_y \in G$. If $P(z_j|g_y)$ is normalised, then $\Psi_4(z_j)$, which is the same as the marginal probability distribution $P(z_j)$.

Thus, using Equations 5.4 and 5.5, the topical relevance weight $w_z(t_i)$ of the term t_i is calculated as follows:

$$w_z(t_i) = \sum_{z_j \in \Gamma_3^{-1}(t_i)} \left\{ P_j(t_i|z_j) \times \sum_{g_y \in \Gamma_4^{-1}(z_j)} P_y(z_j|g_y) \right\} \tag{5.6}$$

5.1.5 Ranked Feature Fusion

The feature fusion stages of the USIF framework operate independently (see Figure 5.1). In the first stage, two modules are integrated (i.e., the intra-cluster topical relevance and the conceptual agglomeration) to select a set of representative topical terms (i.e., lexical features). In Section 5.1.4.1, it was noted that some systems might ask for the top-k representative topical terms rather than specifying the level of abstraction. In such cases, we select all the terms required to describe the nodes in the lowest full-level. We then select the remaining terms from the highest partial-level using a ranking score calculated by $r(t_i)$. In this study, we use $r(t_i) =$

$w_z(t_i)$, which is derived from each cluster rather than the collection. When a parent node in the taxonomy agglomerates children nodes, the score of term t_i in the parent node is calculated by summing up the term scores assigned by $r(t_i)$ from the children nodes.

In the second stage, the selected topical terms, which are ranked based on the aggregated scores from the first stage, are re-weighted using the fused score estimated by the feature fusion module (see Figure 5.1). The module estimates the relevance of each selected topical term in relation to its topical relevance, thematic relevance and its global statistic in the collection. Thus, let the probability of a selected term t_i of topically relevance be $P(t_i|Z)$ and the term of thematically relevance be $P(t_i|G)$. The joint probability is $P(t_i|Z) \times P(t_i|G)$. Additionally, let $df(t_i)$ be the document frequency of t_i. If it assumed that $P(t_i|Z) \propto w_z(t_i)$ and $P(t_i|G) = w_g(t_i)$, we can write $P(t_i|Z,G) \propto w_z(t_i) \times w_g(t_i)$. By using the concept of joint probability, the fused feature score is calculated as follows:

$$w(t_i) = w_z(t_i) \times w_g(t_i) \times df(t_i) \tag{5.7}$$

Thus, if the set $T' = \{t_1, t_2, \ldots, t_k\}$ represents the topical terms that are selected in the first stage, the ranked feature fusion module (see Figure 5.1) then produces the set $T = \{(t_i, w(t_i))|t_i \in T'\}$, which represents the relevant features that describe the user's information needs.

5.1.6 Unsupervised Multi-Fusions Algorithm

Algorithm 3 shows the implementation of the main steps of our proposed USIF framework. Lines 2 to 9 estimate the topical relevance of the selected lexical features, line 8 determines a distribution proportional to marginal probability distribution $P(z_j)$ and line 9 ascertains the summation of $P(t_i|z_j) \times P(z_j)$, which is the estimated topical relevance $w_z(t_i)$ for a term t_i. Lines 11 to 14 show the set of selected topical terms with corresponding fused scores, line 12 checks whether the term $\Omega[i]$ is a selected feature obtained by the integration of our conceptual agglomeration of clusters and $r(t_i)$, line 13 estimates the relative term importance $w[i]$ of term $T'[i]$ and line 14 adds the term $T'[i]$ and its score $w[i]$ as a pair to the set T. Line 15 returns the set T of feature score pairs.

5.1.6.1 Time Complexity Analysis

The proposed USIF framework uses LDA and the BKM algorithm in its feature fusion stages. As the USIF framework was based on our SIF model, it is insensitive to the number of topics

Algorithm 3: USIF algorithm

```
   Input  : A matrix $P_{zg}$ that contains $P(z|g)$, a matrix $P_{tz}$ that contains
            $P(t|z)$, a vector $df$ that contains $df(t)$, a vector $T'$ that contains
            the representative topical terms and a vector $\Omega$ that contains the
            vocabulary terms.
   Output: A set $T$ of relevant features with corresponding scores.
 1 Let $w_z$ be a vector of size $T'$;
 2 for $i = 1$ to $T'$ do
 3     $w_z[i] = 0$;
 4     Let $P_z$ be a vector of size $V$;
 5     for $j = 1$ to $V$ do
 6         $P_z[j] = 0$;
 7         for $k = 1$ to $N$ do
 8             $P_z[j] = P_z[j] + P_{zg}[j][k]$;
 9         $w_z[i] = w_z[i] + P_{tz}[i][j] \times P_z[j]$;
10 Let $T = \emptyset$;
11 for $i = 1$ to $T'$ do
12     if $\Omega[i] \in T'$ then
13         $w[i] = w_z[i] \times df[i]$;
14         $T = T \cup \{(T'[i], w[i])\}$;
15 return $T$;
```

parameter (V) (see Section 6.9.4). Thus, the LDA's time complexity continues to be $\propto \mathcal{O}(|G|)$ for the second stage and $\propto \mathcal{O}(|G_{c_r}| \times |C|)$ for the first stage where $|G_{c_r}|$ is the total number of paragraphs in cluster c_r documents. The time complexity of the BKM algorithm is linear to the $|D^+|$ [Steinbach et al., 2000], which is relatively small in our case. Thus, the time complexity of the BKM for the first stage is $\propto \mathcal{O}(|D^+|)$. However, both algorithms only need to be run once and can be run offline.

Line 1 of Algorithm 3 takes $\mathcal{O}(1)$ basic operations to complete. Lines 2 to 9 take $\mathcal{O}(|Z| \times |T'| \times |G|)$ basic operations to complete. Line 10 takes $\mathcal{O}(1)$ basic operations to complete. Lines 11 to 14 take $\mathcal{O}(|T'|)$ basic operations to complete. Line 15 takes $\mathcal{O}(1)$ basic operations to complete. The total basic operations required by the algorithm are $\mathcal{O}(1) + \mathcal{O}(|T'| \times |Z| \times |G|) + \mathcal{O}(1) + \mathcal{O}(|T'|) + \mathcal{O}(1) \propto \mathcal{O}(|T'| \times |Z| \times |G|)$. Thus, the time complexity of Algorithm 3 is $\mathcal{O}(|T'| \times |Z| \times |G|)$. As the number of topics is usually very small and the performance is not sensitive to the number of topics, the required time complexity is effectively $\mathcal{O}(|T'| \times |G|)$. However, it must be noted that $|T'|$ is small where $1 \leq |T'| \leq k$ and our USIF is not sensitive to the parameter k (see Figure 6.27 [right]).

5.2 The Proposed SSIF Framework

5.2.1 Introduction

As a set of irrelevant documents, negative feedback has been extensively used in many relevance discovery models to enhance the selection and weighting of features that are specifically relevant to what the user needs [Li et al., 2015, 2011, 2012, Tao et al., 2011, Yuefeng and Ning, 2006]. However, using negative feedback is challenging, as these documents are not domain-specific; rather they are topically diverse, skewed and suffer from uncertainties [Li et al., 2011, 2012, Zhong et al., 2012]. Additionally, collecting high-quality negative documents is difficult, expensive and time consuming [Algarni, 2011, Soleimani and Miller, 2016]. As unsupervised relevance discovery models are not discriminative, they cannot deal with the features that frequently appear in both positive and negative feedback [Hou et al., 2010, Man et al., 2009]. Such features are noisy and problematic and may hinder the performance of many IR, IF, DM and ML applications, as these features cannot be used to distinguish between relevant and irrelevant documents. Supervised models are discriminative and developed to consider positive and negative samples in training collections differently [Joachims, 2002, Man et al., 2009, Sebastiani, 2002]. However, supervised models are sensitive to: (1) the feature type they use [Li et al., 2015, 2012]; (2) the uncertainties available in any positive samples [Alharbi et al., 2018a, Li et al., 2017c]; (3) the skewness of one sample compared to another [Li et al., 2017c, Xue and Zhou, 2009]; and (4) the effectiveness of the discrimination algorithm [Man et al., 2009, Yang and Pedersen, 1997]. This study considered whether a method could be developed that combines the advantages of both the supervised and unsupervised learning methods to overcome their limitations.

Of numerous unsupervised fusion techniques, topic-based models are the only models that explicitly assume that a document may contain multiple topics or themes [Blei et al., 2003, Gao et al., 2014b, 2015]. These models, specifically LDA, learn a function from a set of unlabelled documents that describes the hidden topical structures (e.g., latent topics) available in the documents [Blei et al., 2003, John Lu, 2010]. The focus of this learning is to weight features from the detailed composition of the documents in a way that allows the function to generate the hidden structures. Thus, such models can identify distributions of features to summarise specific aspects in documents (e.g., the topics or themes or some essential aspects of meaning) [Blei et al., 2003, Hofmann, 2001]. The feature distribution does not overfit the given documents (or collection) [Blei et al., 2003, Wei and Croft, 2006]. However, the features may not be specific to the topics of interest in the collection, as some features may appear in documents that are not relevant to these topics [Li et al., 2015, 2010]. Additionally, unlike many supervised models, the SVM uses a set of labelled training examples to learn a function that associates new examples with corresponding labels [Joachims, 2002, Man et al., 2009]. The focus of this learning is to select and weight features from the training examples in a way that allows the learned function to separate one label from another. Thus, this learned function can identify the discriminative power of features to separate a given collection of documents from other collections and can be used to select specific features [Joachims, 1998, Sebastiani, 2002]. However, the function cannot address hidden semantic structures to summarise a given collection. Consequently, the SVM model performs poorly for relevant feature discovery as reported in several studies [Algarni and Li, 2013, Gao et al., 2015, Li et al., 2015, 2011, 2010, Zhong et al., 2012].

This section presents our innovative and highly-effective SSIF framework [3]. This framework discovers specifically relevant topical terms that reflect users' information preferences. The framework integrates supervised and unsupervised algorithms to select and then weight these topical terms at two independent stages of feature fusion. Like our USIF framework, the SSIF framework also adheres to the same multiple fusion strategy in its stages and the fusion modules are also developed based on our SIF and UR models. In the first stage (see Figure 5.5), the SSIF framework selects a set of representative, weighted topical terms using the discriminative SVM algorithm incorporated with the adapted UR method. This stage ensures that the selected terms are specifically relevant to what the user needs, as the SVM requires that both relevant and irrelevant documents and the available uncertainties in the relevant documents be considered before applying the SVM. In the second stage, the SSIF framework estimates the informativeness of the selected specific terms from the first stage via the integration of their topical and thematic relevance and their global exhaustivity in the collection of relevant documents. As users are normally interested in relevant documents, the framework uses the estimated relevance from the second stage to re-weight (i.e., scale) the selected weighted terms of the first stage. The experimental results, presented in Section 6.8.5 and discussed in Section 6.9.5, show that our SSIF framework is more highly and significantly effective than both popular and state-of-the-art baseline models despite the features they fuse, how they fuse them or even the learning or mining algorithms that generate these features.

[3] 'SSIF' stands for **S**upervised **S**election of **I**nformative **F**eatures.

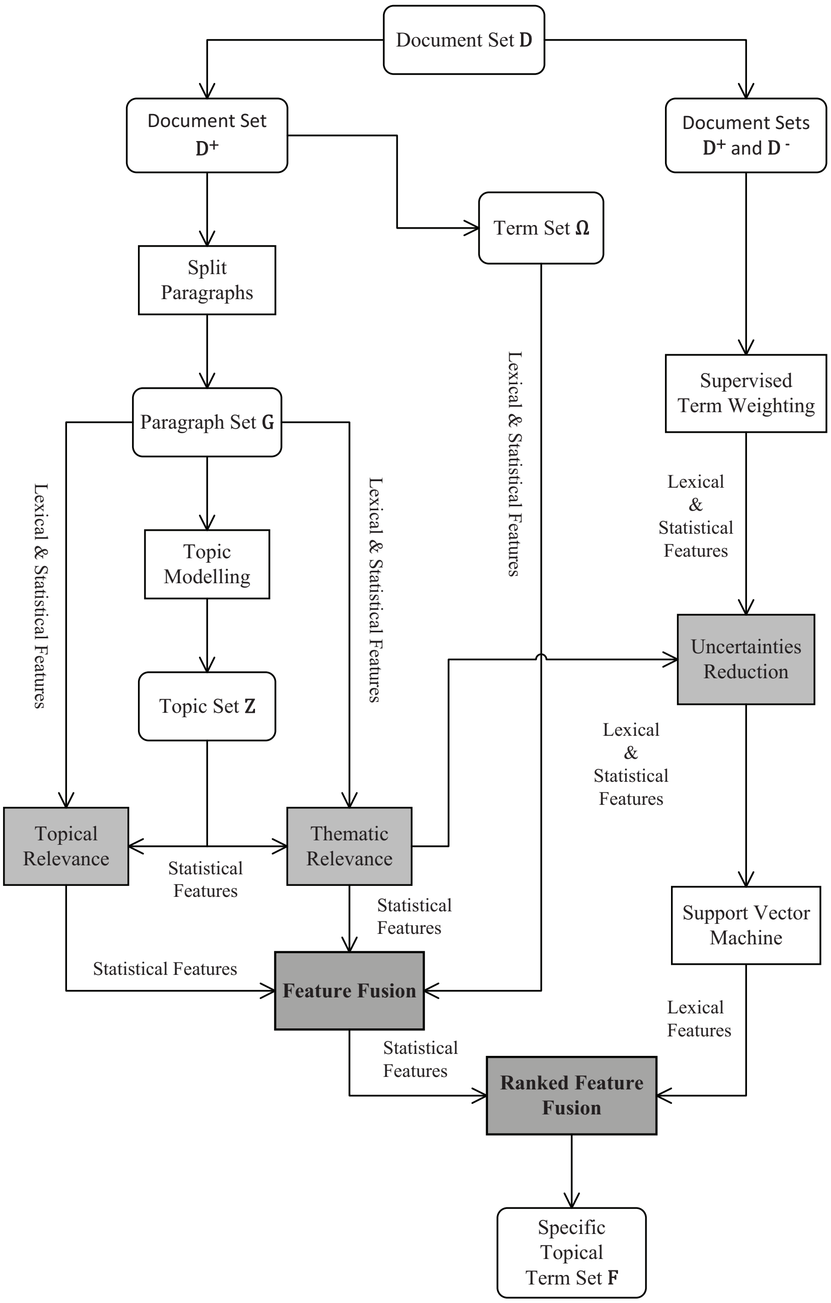


Figure 5.5: The structure of the SSIF framework.

As Figure 5.5 shows, the proposed SSIF framework uses both positive and negative training documents for feature selection in the first stage and uses only relevant documents for feature weighting in the second stage. As noted above, the framework extends multiple random sets to model the complex relationships between different entities in the relevant collection and thus estimates a more accurate relevance score for topical terms. As Figure 5.5 shows, the used entities are the collection's terms, paragraphs and the latent topics in the paragraphs. Figure 5.5 also shows the flow of the features (lexical and statistical) between the fusion modules.

The following sections describe our SSIF framework in detail. First, Section 5.2.2 outlines the problem formulation. Next, Section 5.2.3 provides details of the fusion stages of the framework. Section 5.2.4 then outlines how our SSIF framework integrates between the outputs of each stage. Finally, Section 5.2.5 describes the SSIF algorithm and its time complexity.

5.2.2 Problem Formulation

Assume that a user maintains a collection of news stories D for research purposes. The collection contains a set of documents that are related to some ‘economic espionage’ scenarios that have occurred around the world. However, the user is only interested in some forms of espionage. Thus, the user decides to split the collection D into a relevant (i.e., positive) collection D^+ and an irrelevant (i.e., negative) collection D^-. The relevant documents in collection D^+ discuss the topics of the scenarios in which the user is interested, such as ‘industry espionage’, ‘technical espionage’, ‘commercial espionage’ and ‘corporate espionage’. The user keeps irrelevant news documents in the D^- collection that discuss unwanted topics such as ‘military espionage’ and ‘political espionage’. To enrich D^+ and remain abreast of new scenarios of economic espionage, the user needs to collect more news documents from the Internet that are pertinent to the topics of interest in D^+. To achieve this goal, the user needs a framework for selecting and weighting features to describe the collection effectively. The weighted features will be used to gather the relevant documents.

Based on the above example, it is likely that there will be many shared features between relevant and irrelevant topics of interest in both the D^+ and D^- documents. However, given that the user is only interested in the topics of D^+, it requires more emphases on the relevant information that comes from D^+ documents. The irrelevant information available in the D^- documents is also useful and needs special treatment. Thus, unlike traditional, supervised, relevant feature discovery models, the proposed SSIF framework follows the approach of our USIF framework by treating feature selection and feature weight as two independent feature fusion tasks. The SSIF framework integrates three crucial characteristics (see below) of important features to ensure effective fusion and thus accurately selects and weights the topical terms to effectively represent the user’s information needs.

- **Feature Specificity**: Selecting features that can discriminate between the D^+ collection and the D^- collection is critical [Maxwell and Croft, 2013]. Fang et al. [Fang et al., 2004] argue that a new document that has more occurrences of specific features should be favoured as relevant to a given corpus. Thus, the features should be specific to the given collection. We argue that a supervised learning algorithm is effective for selecting features that are specific to the collection. Some examples of potential supervised learning methods are SVM [Joachims, 1998], BM25 [Robertson and Zaragoza, 2009] and RFD$_2$ [Li et al., 2015].

- **Feature Informativeness**: The features should represent the essential aspects of meanings of the user's information needs. If the informativeness of a feature is increased, then the chance of a document matching the feature being relevant to the user's information needs is increased. Thus, informativeness should increase precision.
- **Feature Exhaustivity**: The features should be exhaustive [Yuefeng and Ning, 2006] of the user's information needs. It should be noted that the exhaustivity of a feature refers to the coverage of various subjects of the user's information needs. If the exhaustivity of a feature is increased, then the chance of the feature matching a relevant document is increased. Thus, exhaustivity should increase the recall by reducing the chance of dropout of a relevant document.

The remaining problem relates to determining how to accurately estimate these three aspects and integrate them. This research showed that a set of features first need to be found that are specific to a user's information needs. Next, the relevance of these specific features needs to be jointly estimated from their informativeness and exhaustiveness. This research incorporates supervised (i.e., BM25 and SVM) and unsupervised (i.e., LDA) learning algorithms to determine the specific features and uses both the topical relevance and thematic relevance of a feature to estimate its informativeness.

5.2.3 SSIF Fusion Stages

Similar to the USIF framework, under the SSIF framework, the tasks of feature selection and feature weighting are undertaken independently at two different fusion stages. However, unlike under the USIF model, the SSIF framework treats the selection task as a supervised problem during the first stage to identify those features that are specific to the relevant topics of interest in the D^+ collection, but not those that are irrelevant and captured by the D^- collection. Additionally, similar the USIF framework, as a user can only be interested in relevant documents, the SSIF framework views the weighting task in the second stage to be an unsupervised problem. It uses the positive collection D^+ to estimate a more accurate weight for each specific feature selected from the first stage. However, due to the uncertainties available in positive documents, the hugely diverse topics in the negative documents and the large number of common features between both positive and negative documents, the selection and the weighting problems are challenging. The next two sections examine the feature fusion stages of the proposed SSIF framework.

5.2.3.1 Stage 1: Selecting Specific Topical Terms

This research uses the supervised learning SVM to select specific features (i.e., topical terms that are related to the topics of interest). However, the SVM is a term-based model and does not consider any latent topical structure in either positive or negative training documents. Thus, at the first stage, to add a topical representation in an implicit manner, the SVM is integrated with the adapted version of our UR method (see Section 5.1.4.2). This integration sought to reduce the uncertainties in D^+ documents, as they discuss the topics of interest for the user and thus assist the SVM to learn a more accurate hyperplane (as described in the following section). To do this, the SVM has to first be trained. Different types of initial feature weights are used to represent the training documents (e.g., IDF, TFIDF, BM25 and etc.). From those weights, the BM25 is combined with our UR method, as it is supervised and performed the best in our UR experiments (see Section 6.8.3 and our study published in [Alharbi et al., 2018a]). The following section gives a brief description of the SVM.

Support Vector Machine

A SVM is a supervised classifier that is theoretically defined by a hyperplane that separates relevant and irrelevant documents of a class. First, the classifier learns a hyperplane from a set of training examples. Then, the learned hyperplane is used to categorise new examples based on which side of the hyperplane a new example sits. To train the SVM to classify documents, each document is represented with a list of term weight pairs. Each term in the list is a unique term in the document and the corresponding weight attributed to that term represents its significance in the document. The equation of a hyperplane is $\overline{\beta} \cdot \overline{x} + c = 0$, where $\overline{\beta}$ is the weight vector, $\overline{x}$ is the term vector and c is a constant. In the learning process, the SVM sets $\overline{\beta}$ and c. This allows the hyperplane to optimally separate the positive (relevant) examples and negative (irrelevant) examples. The distance of each example document from the hyperplane is positional $\overline{\beta}$. Such that each element of $\overline{\beta}$ is proportional to the distance of the corresponding term from the hyperplane. The main output of the SVM are $\overline{x}$ (i.e., the vector of terms in the vocabulary) and $\overline{\beta}$ (i.e., the distance vector of the corresponding terms in $\overline{x}$)).

5.2.3.2 Stage 2: Weighting Specific Topical Terms

The theoretical approach used in the second stage of the USIF framework was adopted, as this approach has been shown to effectively and accurately estimate the relevance of the selected features. Thus, at the second stage of the proposed SSIF framework, the topical and thematic significances of a specific feature, which are estimated from the relevant collection D^+, are used jointly. The estimated significances are also combined with the document frequency, as this is the best global statistic to efficiently and effectively indicates the exhaustivity of relevant terms. The topical significance (i.e., relevance) of a specific term t_i to the hidden topics that are discussed is D^+ (i.e., $P(t_i|Z)$) is estimated based on Equation 5.6, as theoretically justified in Section 5.1.4.2. Similarly, the thematic relevance of the term t_i (i.e., $P(t_i|G)$), which is selected at the first stage, is estimated using Equation 5.3. Both significances of the term t_i are probabilistically combined (i.e., $P(t_i|Z) \times P(t_i|G)$) to estimate t_i informativeness globally at the collection D^+ level (i.e., $P(t_i|Z, G)$).

5.2.4 Ranked Feature Fusion

Both the proposed SSIF and USIF frameworks conduct the selection and weighting of topical terms as independent tasks at two separate stages. However, unlike the USIF framework, the SSIF framework treats the selection task as a supervised problem to select those terms that are specific to the topics of interest in the relevant collection D^+ using the negative documents of D^-. In relation to feature weighting, the SSIF framework adopts the same approach as that adopted by the USIF framework and treats this task as an unsupervised problem in which only D^+ is used. One issue that remained was to determine how to integrate the features produced at each stage without losing any important information. As noted in Section 5.2.2, to create an effective fusion between the estimated features of the two stages and thus effectively select and weight relevant features that are specific to what a user needs, the ranked feature fusion module (see Figure 5.5), incorporates the three previously identified characteristics (see below).

- **Specific**: The top-k ranked features (i.e., the topical terms sorted in descending order) are selected from the integration between BM25, our adapted UR method and the SVM at the first stage. Let $d_x = \{t_1, t_2, t_3, \ldots, t_k\}$. Before training the SVM, each document

$d_x \in D$ is scored using the combination of BM25 and the UR method, such as $d_x = \{(t_i, bm25(t_i) \times ur(t_i))|t_i \in \Omega\}$ where $ur(t_i) = w_g(t_i)$ in this study. This combination sought to reduce the uncertainties in the relevant topical terms before the application of the SVM (see the justification for this approach in Section 4.2.4). After training the SVM, the SVM provides two vectors: $\overline{x}$ (i.e., the vector of terms in the vocabulary) and $\overline{\beta}$ (i.e., the distance vector of the corresponding terms in $\overline{x}$). Let $H = \{(t_i, w_s(t_i))|t_i \in \overline{x}\ \&\ w_s(t_i) \in \overline{\beta}\}$. H is sorted in descending order of $w_s(t_i)$ and the set $F \subseteq H$ of the top-k terms is taken as specific topical terms.

- **Informative**: The integration of the topical and thematic relevance modules (see Figure 5.5) is used to represent the informativeness of the selected features. As the user's main interests are located in the D^+ documents, these two modules fuse different topical features to estimate the significance of the terms in D^+ to the hidden topics and themes. Thus, given a specific term t_i , its informativeness is estimated as the joint probability of its topical relevance $P(t_i|Z)$ with its thematic relevance $P(t_i|G)$. In this study, as we did in our study of the USIF framework, it was assumed that $P(t_i|Z) \propto w_z(t_i)$ and $P(t_i|G) = w_g(t_i)$. Their joint probability was written as $P(t_i|Z, G) \propto w_z(t_i) \times w_g(t_i)$.

- **Exhaustive**: Global frequency is a strong indicator of term importance at the collection level [Bendersky and Croft, 2012] and can be used to optimise feature weights [Bendersky and Croft, 2008, Xue et al., 2010]. The global frequency (or global statistic) of a feature is defined as its frequency across all documents in the collection. This feature indicates which portion of a collection (i.e., how many documents) is covered by a given feature (e.g., a term). This research uses document frequency as the estimation of the exhaustivity of specific terms selected in the first stage.

The next issue is to determine how the above three characteristics can be accurately fused in such a way that the specificity of the selected features would not be compromised by their informativeness and exhaustivity or vice versa. Thus, in this research, the top-k specific features set F have their weights scaled by the linear combination of the features' topical, thematic and global significances in D^+ (previously estimated by Equation 5.7, as $F = \{(t_i, w_s(t_i) \times w(t_i))|(t_i, \_) \in H\}$).

5.2.5 Supervised Multi-Fusions Algorithm

Algorithm 4 shows the main implementation steps of the proposed SSIF framework. The algorithm is similar to that of the USIF framework, especially in estimating the topical, thematic and global relevance of the selected specific features (lines 1 to 13). Additionally, as in the USIF framework's algorithm, the details of applying LDA to the paragraph set G to generate the topic set Z and the calculations of the required probabilities are omitted, as they can be learned from the SIF and SIF2 algorithms (see Sections 3.3.4 and 4.1.6). However, two vectors F' and F'' are defined to store the top-k features produced by the first stage and their corresponding weights, respectively. Notably, Line 14 shows how the features from both the SSIF framework's stages can be effectively combined to maintain the specificity, informativeness and exhaustivity of features.

Algorithm 4: SSIF algorithm

```
Input  : A matrix $P_{zg}$ that contains $P(z|g)$, a matrix $P_{tz}$ that contains
         $P(t|z)$, a vector $df$ that contains $df(t)$, a vector $F'$ that
         contains the top-$k$ terms of the SVM, a vector $F''$ that
         contains the corresponding weights of the SVM terms in $F'$
         and a vector $\Omega$ that contains the vocabulary terms.
Output: A set $F$ of features with corresponding scaled scores.
1  Let $w_z$ be a vector of size $F'$;
2  for $i = 1$ to $F'$ do
3      $w_z[i] = 0$;
4      Let $P_z$ be a vector of size $V$;
5      for $j = 1$ to $V$ do
6          $P_z[j] = 0$;
7          for $k = 1$ to $N$ do
8              $P_z[j] = P_z[j] + P_{zg}[j][k]$;
9          $w_z[i] = w_z[i] + P_{tz}[i][j] \times P_z[j]$;
10 Let $F = \emptyset$;
11 for $i = 1$ to $F'$ do
12     if $\Omega[i] \in F'$ then
13         $w[i] = w_z[i] \times df[i]$;
14         $F = F \cup \{(F'[i], F''[i] \times w[i])\}$;
15 return $F$;
```

5.2.5.1 Time Complexity Analysis

The time complexity of Algorithm 4 is similar that of the USIF framework's algorithm (i.e., $\propto \mathcal{O}(|F'| \times |G|)$), as it linearly depends on the size of G and F'. However, it should be noted that T' and F' in the two algorithms are relatively small in size and depend on the k parameter.

As both frameworks were developed based on our SIF theory, both frameworks inherited its insensitivity to the k parameter (see the experimental evaluation chapter). Further, unlike the stages of the USIF framework where LDA applied $|C|$ times in the first stage and once in the second stage, our SSIF framework only needs to apply LDA once, while the generated topics are used across the two stages. The time complexity of the LDA remains the same (i.e., $\propto \mathcal{O}(|G|)$), the only difference is the use of the SVM in the first stage of the SSIF framework that requires a polynomial computational time that depends on the training instances [Man et al., 2009]. However, both the SVM and LDA were only required to be run once and were run offline in our experiments.

5.3 Chapter Summary

This chapter introduced two innovative and highly-effective frameworks that can be used to discover relevant features that describe user information preferences. Unlike conventional relevance discovery models, the proposed frameworks treat feature selection and weighting as two independent tasks. Over two different stages, the frameworks first identify a representative set of topical terms and then re-estimate their informativeness using a complex integration of multiple learning algorithms and fusion-based models. The integration is managed by multiple ERSs based on the theoretical merits of our SIF and UR models (as described earlier in this thesis).

The proposed unsupervised USIF framework elegantly addresses the challenges that arise in selecting representative terms from an unbalanced set of topics discussed in a small collection of relevant documents that describe a user's information needs. In the first stage, a conceptual agglomeration technique was developed that is based on the fusion of lexical and statistical features that are discovered via the integration of document clustering and topic modelling algorithms. An agglomeration technique was used to select a predetermined set of inter-cluster, topical terms from unbalanced but equally relevant clusters of documents. As traditional clustering algorithms do not consider the multi-topic structure of documents, the identification of relevant, intra-cluster topical terms is difficult. To address this issue, our USIF framework employed the SIF model to estimate the topical relevance of such terms. In the second stage, the relevance of the selected terms was re-estimated based on fusions of their topical, thematic and global significance (as measured by our SIF model and an adapted version of the UR method) at the collection level rather than at the unbalanced-clusters level. The experimental results (see

Section 6.8.4) demonstrate the superiority of the performance of the USIF framework in IF and RRT over both supervised and unsupervised state-of-the-art baseline models. The experimental results also confirm the merits of the proposed framework in which the problems of feature selection and feature weighting can be addressed independently. The results also show how topic modelling, document-clustering and multiple fusion-based models can be integrated in an unsupervised way to discover relevant features that occur unevenly across the unbalanced topics that appear in a collection of long documents.

Similarly, the proposed supervised SSIF framework sophisticatedly and effectively addressed the difficulties that arose in discovering topical terms that are specifically relevant to a user's needs based on small samples of positive and negative documents. The SSIF framework is similar to the USIF framework; however, it conducted the selection and weighting of topical terms that frequently appear in both positive and negative topics of interest over independent stages differently. In the first stage, the selection problem was addressed via the fusion of supervised (i.e., SVM) and unsupervised (i.e., LDA) learning algorithms in which the inherited uncertainties in positive documents were addressed using the adapted UR method. Second, the proposed framework learned a more accurate weight for specific topical terms, which were selected during the first stage, via an unsupervised integration of multiple fusion-based models that was managed by the ERSs theory of our SIF model and an adapted UR method. The weighting problem was addressed by determining the joint estimation of the topical and thematic relevance of the selected terms in the positive documents and their global exhaustivity across these documents. The experimental results (see Section 6.8.5) showed that our SSIF framework is highly effective and significantly outperformed all the baseline models in all performance measures across both IF and RRT tasks. This study developed a promising methodology that combines the advantages of supervised and unsupervised learning for feature selection and effectively uses the topical features and global statistics of low-level terms for feature weighting.

In the next chapter, an experimental evaluation is undertaken of all the proposed models and frameworks in this thesis based on the widely accepted IF-based system methodology. The proposed techniques are also evaluated in relation to the ranking of relevant features that were manually identified by NIST's domain experts. Fifty collections of long documents from the popular RCV1 dataset are used for the evaluation purposes, including seven standard performance measures, TREC filtering topics and more than 20 different baseline models. Additional details about the experiments are provided in the next chapter.

Chapter 6

Experimental Evaluation

6.1 Introduction

As mentioned in Chapter 1, the research in this thesis proposes several TFS techniques for relevance discovery. These techniques deal with uncertainties in the relevant documents that describe user information needs using data fusion approaches. For example, the SIF model fuses different features from relevant documents to discover informative topical terms on a global level. The SIF2 model revises SIF model and solves the generalised weight hypothesis of topical terms that SIF was developed upon to tackle the nonmonotonic problem of some relevant features. The UR method reduces uncertainties in relevant features discovered by existing TFS techniques by fusing different features to estimate the passage-level evidence of relevance. Two other fusion-based frameworks, namely, USIF and SSIF are proposed to deal with the bias toward frequent topics and the features that appear in both relevant and irrelevant documents, respectively.

This chapter presents and thoroughly describes the experimental evaluation methods for the proposed TFS models and frameworks. The chapter describes the essential aspects of the experimental evaluation, including the evaluation hypotheses, experimental design, data collections, performance measures, baseline models and their experimental settings. Then, the results are presented, discussed and analysed separately for each model and framework based on their evaluation hypotheses. The popular RCV1 is selected as the benchmark dataset including its TREC-11 topics for IF tasks. Seven standard evaluation metrics are used to measure different aspects of the effectiveness of the performance of the proposed models and frameworks in IF and RRT applications. Also, the standard paired t-test (aka Student's t-test) is used to test how significant the difference is between the results of the proposed techniques and the baseline models in both the IF task as well as in ranking relevant features that are identified by TREC's domain experts. A variety of state-of-the-art and popular baseline models are selected, and their results are compared with the proposed models and frameworks. These baseline models use different fusion strategies, text features and mining and learning algorithms.

6.2 Hypothesis

Several hypotheses were designed to verify the proposed TFS models and frameworks for discovering relevant features that describe user information needs. In this thesis, each hypothesis was developed to validate the main aspects of a particular model or framework. These hypotheses are presented as follows:

- *Hypothesis 1*: The proposed SIF model can effectively select informative topical terms from a set of relevant documents through the hybrid fusion of different global features discovered by topic modelling and a collection statistic.

- *Hypothesis 2*: The proposed SIF2 model can effectively select the most informative topical terms learned from a collection of relevant documents via the hybrid fusion of different local and global features learned from topic modelling and collection statistics.

- *Hypothesis 3*: The proposed UR method can effectively reduce uncertainties in relevant features through the estimation of the relevance of paragraphs that can be used to re-estimate the relevance of features (i.e., terms) discovered by existing TFS techniques.

- *Hypothesis 4*: The proposed USIF framework can effectively select and re-weight topical terms that occur in clusters of relevant documents that contain frequent topics and less frequent but equally important ones via the hybrid fusions of different features discovered by topic modelling, document clustering and global statistics.

- *Hypothesis 5*: The proposed SSIF framework can effectively select and re-weight relevant topical terms that frequently appear in relevant and non-relevant training documents through the hybrid fusions of different features learned from the same documents by a combination of supervised and unsupervised algorithms as well as global statistics.

In the following sections, each hypothesis will be experimentally evaluated using the standard and widely accepted IF system-based methodology similar to the studies in [Bashar and Li, 2018, Bashar et al., 2017, Gao et al., 2015, Li et al., 2015, Wu et al., 2006, Zhong et al., 2012]. In addition to the IF-based methodology, the set of relevant terms identified by NIST domain experts will be used to evaluate those topical terms discovered and re-ranked by the proposed TFS models and frameworks.

6.3 Data Collection

Many published and publicly available datasets have been used in the field of text classification, IR and IF. Among the most popular ones, especially those used by TREC, are the standard Reuters datasets. The Reuters Corpus Volume 1 (RCV1) [Lewis et al., 2004] is selected for all the experiments in this chapter. In the following section, more details about the RCV1 dataset are given.

6.3.1 RCV1

RCV1 consists of 100 collections of documents that cover a wide range of subjects to suit different interests. The first 50 collections, from Collection 101 to 150, are used in this research due to their reliability and high quality as they were manually assessed by domain experts at NIST for TREC[1] in their filtering track [Robertson and Soboroff, 2002, Soboroff and Robertson, 2003]. These collections are usually known as the assessors topics in that track because they were assessed and labelled by human domain experts. However, in this research and to differentiate between an LDA latent topic and the TREC topic, each assessor topic was called a collection. According to Buckley and Voorhees [Buckley and Voorhees, 2000] and other experimental studies in [Gao et al., 2015, Li et al., 2015, 2012, Zhong et al., 2012], this number of collections (i.e., the 50 collections) is sufficient and stable for better and reliable experiments. The last 50 collections (aka intersection topics) were completely labelled by a machine learning algorithm. Thus, they are less in terms of quality and reliability [Li et al., 2012, Robertson and Soboroff, 2002, Soboroff and Robertson, 2003]. Each collection D of the RCV1 has been split into training and testing sets, and each set has some relevant (aka positive) D^+ and irrelevant (aka negative) D^- documents to the topic they describe as illustrated in Figure 6.1.

RCV1 is a large dataset with more than 806,000 documents that are distributed over the 100 different collections. Each document is a news story written by a journalist in English and published by Reuters. Table 6.1 shows the main statistics of the RCV1 dataset while Figure 1.2 illustrates the topic's description of Collection 101 as prepared by TREC's assessors. Moreover,

[1] http://trec.nist.gov/

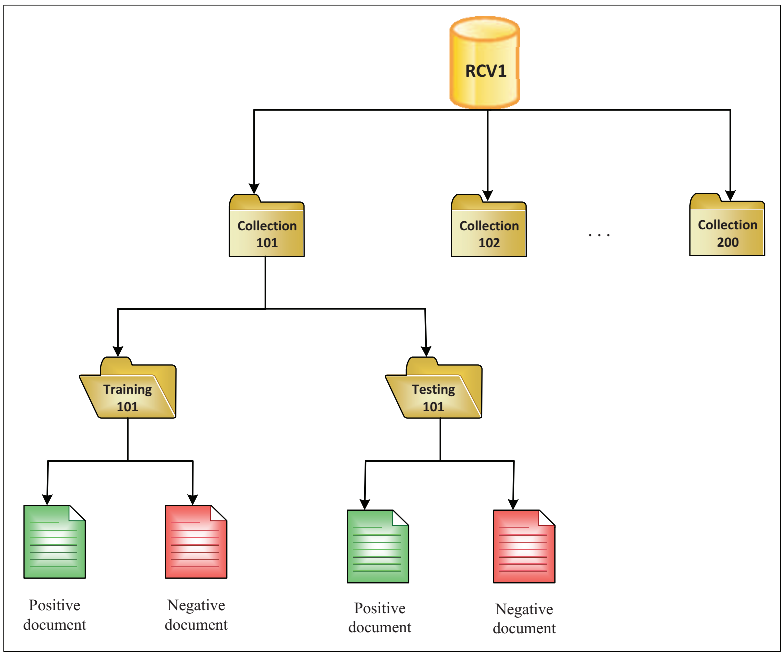


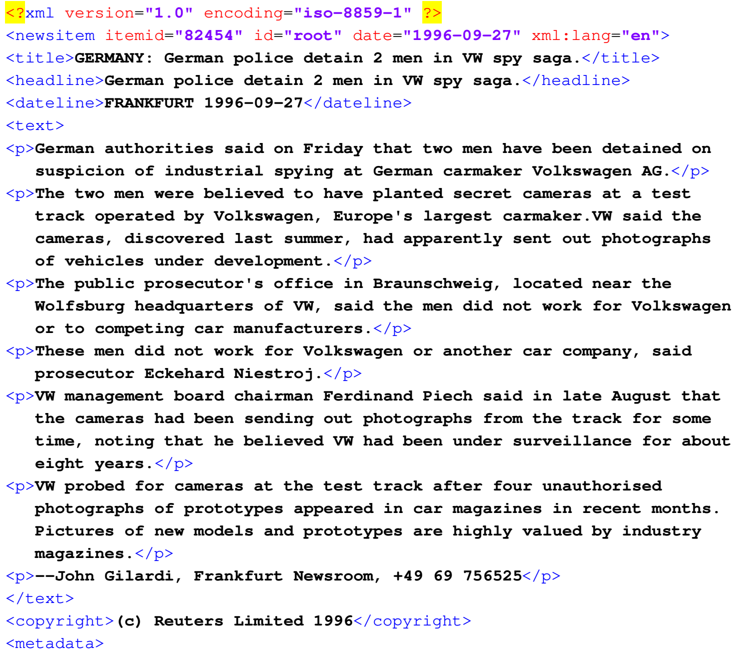
Figure 6.1: The structure of RCV1 dataset.

each document in the RCV1 is in an XML format that has many elements as shown in Figure 6.2. The proposed models and frameworks including the baselines use only the *'title'* and *'text'* elements during the training and testing phases. Each element (i.e., '<title>' and '<p>') is considered a separate paragraph to be used in training the SIF, SIF2, UR models and some parts of the USIF and SSIF frameworks. Thus, each RCV1 document has at least two paragraphs, the '<title>' and at least one content paragraph as a sub-element of the '<text>' element. To eliminate bias in our experiments, all meta-data elements have been ignored. Also, each and every paragraph of the relevant documents are separately split and indexed to facilitate the extraction of smaller sub-topics using LDA as sub-documents (i.e., passages or paragraphs) show better results in IR [Krikon and Kurland, 2011, Xi et al., 2001].

Table 6.1: The main statistics of the RCV1 dataset [Lewis et al., 2004]

Statistic	Value
The total number of documents	806,791
The total number of paragraphs	9,822,391
The total number of terms	96,969,056
The vocabulary size	391,523
The average vocabulary size in a document	75.7
The average document length	123.9

Moreover, each document in the RCV1 dataset is a long document with an average number



```
<?xml version="1.0" encoding="iso-8859-1" ?>
<newsitem itemid="82454" id="root" date="1996-09-27" xml:lang="en">
<title>GERMANY: German police detain 2 men in VW spy saga.</title>
<headline>German police detain 2 men in VW spy saga.</headline>
<dateline>FRANKFURT 1996-09-27</dateline>
<text>
<p>German authorities said on Friday that two men have been detained on
   suspicion of industrial spying at German carmaker Volkswagen AG.</p>
<p>The two men were believed to have planted secret cameras at a test
   track operated by Volkswagen, Europe's largest carmaker.VW said the
   cameras, discovered last summer, had apparently sent out photographs
   of vehicles under development.</p>
<p>The public prosecutor's office in Braunschweig, located near the
   Wolfsburg headquarters of VW, said the men did not work for Volkswagen
   or to competing car manufacturers.</p>
<p>These men did not work for Volkswagen or another car company, said
   prosecutor Eckehard Niestroj.</p>
<p>VW management board chairman Ferdinand Piech said in late August that
   the cameras had been sending out photographs from the track for some
   time, noting that he believed VW had been under surveillance for about
   eight years.</p>
<p>VW probed for cameras at the test track after four unauthorised
   photographs of prototypes appeared in car magazines in recent months.
   Pictures of new models and prototypes are highly valued by industry
   magazines.</p>
<p>--John Gilardi, Frankfurt Newsroom, +49 69 756525</p>
</text>
<copyright>(c) Reuters Limited 1996</copyright>
<metadata>
```

Figure 6.2: A sample of an XML document from collection 101 of the RCV1 dataset.

of more than 12 paragraphs. Figures 6.3 and 6.4 show the paragraphs distributions in the RCV1 training sets used with the experiments of the unsupervised and supervised TFS models and frameworks, respectively. These figures illustrate the suitability of RCV1 documents for topic modelling as each document can discuss multiple topics or sub-topics (i.e., themes) across its paragraphs. Also, these multi-paragraph documents allow LDA to be applied at the paragraph-level as each paragraph contains enough information to extract some topics from, as illustrated in Figure 6.2, and facilitate a more practical usage of the generated latent topics, as shown in Chapter 4.

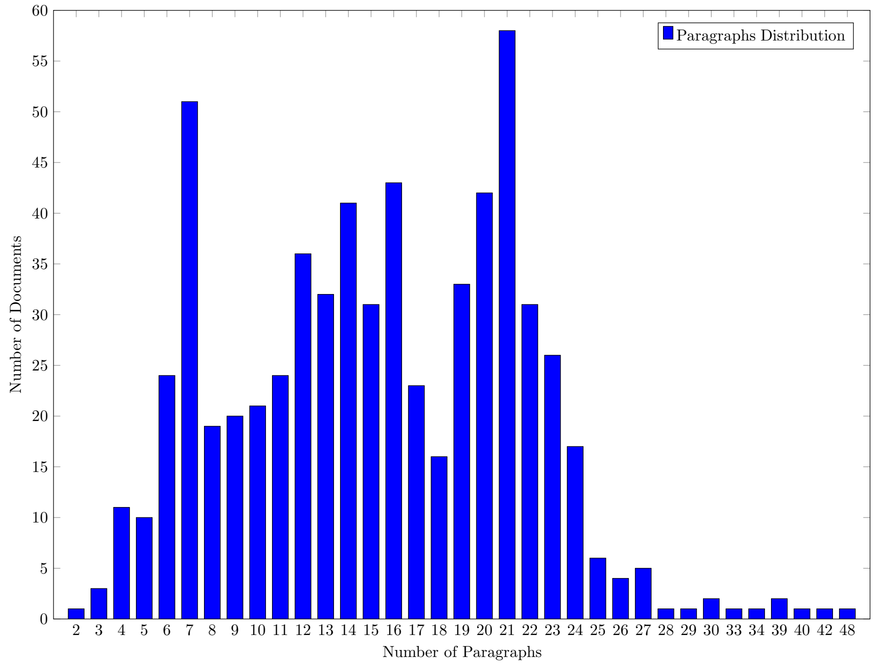


Figure 6.3: The distribution of paragraphs in positive training documents of the first 50 collections of the RCV1 dataset that are used by all unsupervised TFS models and frameworks, including the selected baseline models.

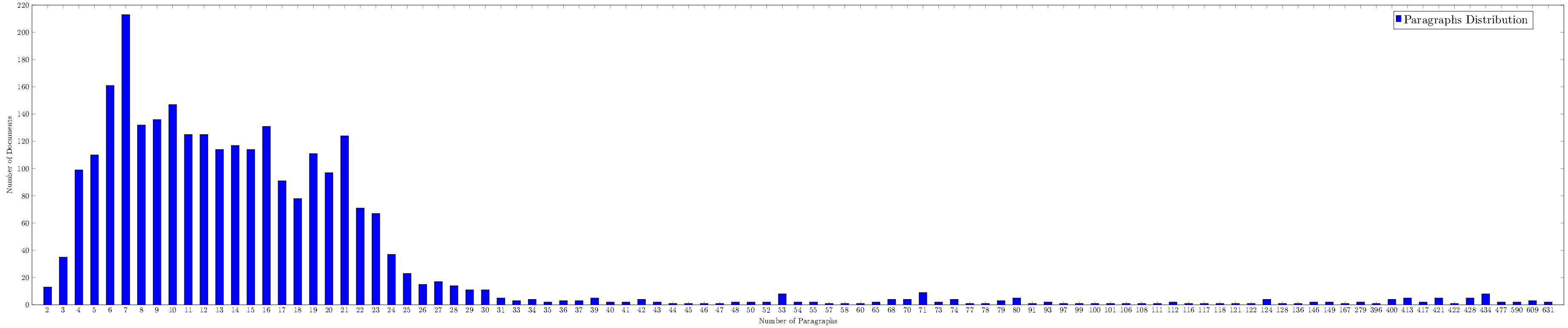


Figure 6.4: The distribution of paragraphs in positive and negative training documents of the first 50 collections of the RCV1 dataset that are used by all supervised TFS models and frameworks, including the selected baseline models.

Table 6.2 shows a statistical summary of the first 50 collections of the RCV1 dataset. Only positive training documents are used in the experiments of the unsupervised TFS models and frameworks with a total number of 639 documents. This number is spread across the 50 collections with an average of fewer than 13 training documents in each collection. This makes most documents exist in the testing sets rather in the training sets as shown in Figure 6.5. Despite the low number of training samples, the proposed techniques maintain higher and robust performance compared to the used baseline models. Supervised TFS algorithms, on the other hand, including the proposed SSIF framework, use both positive and negative training documents in the 50 collections with an average of fewer than 55 documents in each collection compared to more than 377 documents in each testing set, which still makes the testing set much larger in number of documents than the training one as illustrated in Figure 6.6.

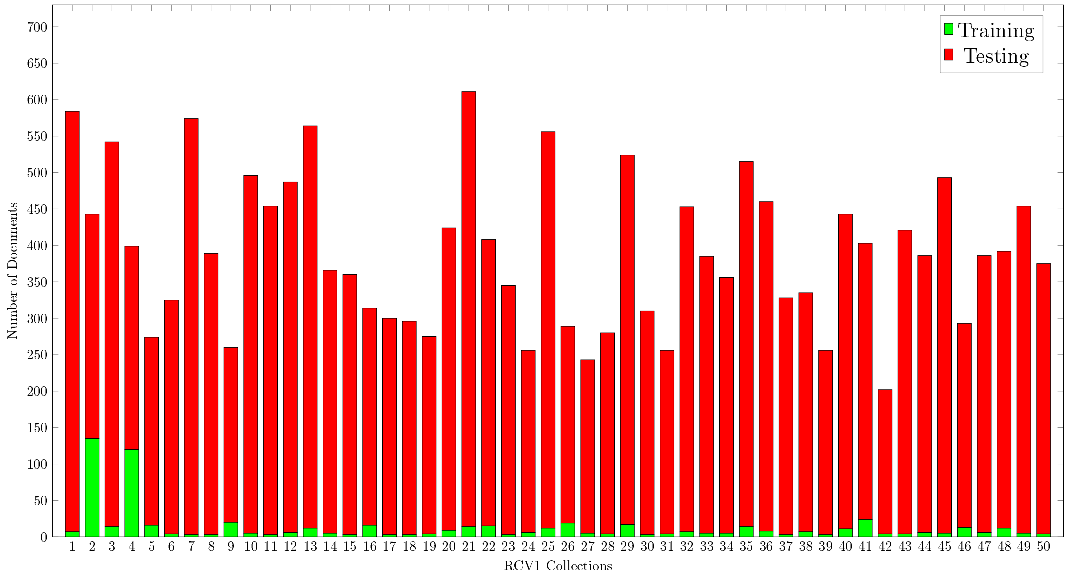


Figure 6.5: The number of training documents compared to the testing documents in the first 50 collections of the RCV1 dataset that are used in the experiments of all unsupervised TFS models and frameworks, including SIF, SIF2, UR and USIF.

6.3.1.0.1 Document Preprocessing Steps

Few preprocessing steps were performed on all RCV1 documents and TREC topics titles during the training and testing phases of the proposed models and frameworks including the baselines. First, all meta-data and stop-words were removed. Second, all keywords were stemmed using the Porter Suffix Stripping algorithm [Porter, 1980]. These preprocessing steps are illustrated in Figure 6.7 and the list of stop-words can be found in Appendix G.

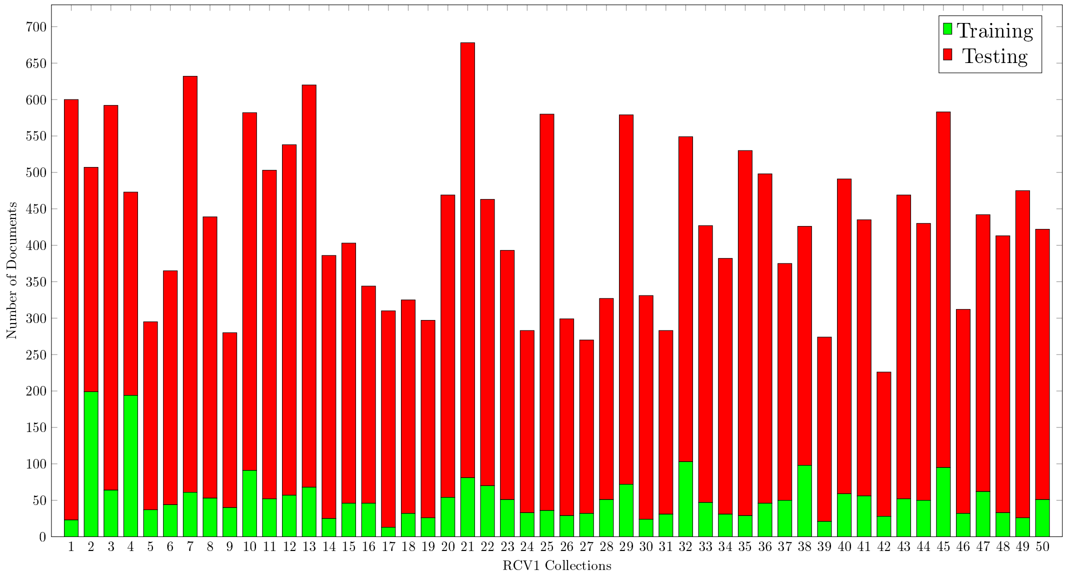


Figure 6.6: The number of training documents compared to the testing documents in the first 50 collections of the RCV1 dataset that are used in the experiments of the SSIF framework and other supervised TFS baseline models.

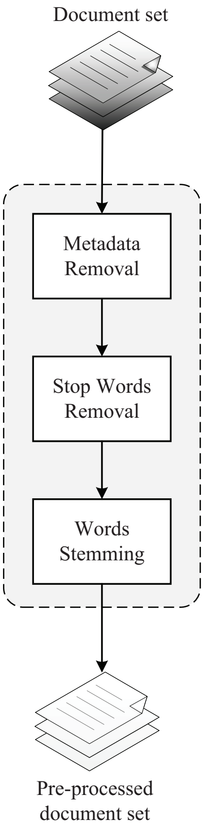


Figure 6.7: The preprocessing steps for all RCV1 documents.

Table 6.2: The statistics of the training and testing sets of the RCV1 dataset

	Training Set			Testing Set		
Collection#	$\|D\|$	$\|D^+\|$	$\|D^-\|$	$\|D\|$	$\|D^+\|$	$\|D^-\|$
101	23	7	16	577	307	270
102	199	135	64	308	159	149
103	64	14	50	528	61	467
104	194	120	74	279	94	185
105	37	16	21	258	50	208
106	44	4	40	321	31	290
107	61	3	58	571	37	534
108	53	3	50	386	15	371
109	40	20	20	240	74	166
110	91	5	86	491	31	460
111	52	3	49	451	15	436
112	57	6	51	481	20	461
113	68	12	56	552	70	482
114	25	5	20	361	62	299
115	46	3	43	357	63	294
116	46	16	30	298	87	211
117	13	3	10	297	32	265
118	32	3	29	293	14	279
119	26	4	22	271	40	231
120	54	9	45	415	158	257
121	81	14	67	597	84	513
122	70	15	55	393	51	342
123	51	3	48	342	17	325
124	33	6	27	250	33	217
125	36	12	24	544	132	412
126	29	19	10	270	172	98
127	32	5	27	238	42	196
128	51	4	47	276	33	243
129	72	17	55	507	57	450
130	24	3	21	307	16	291
131	31	4	27	252	74	178
132	103	7	96	446	22	424
133	47	5	42	380	28	352
134	31	5	26	351	67	284
135	29	14	15	501	337	164
136	46	8	38	452	67	385
137	50	3	47	325	9	316
138	98	7	91	328	44	284
139	21	3	18	253	17	236
140	59	11	48	432	67	365
141	56	24	32	379	82	297
142	28	4	24	198	24	174
143	52	4	48	417	23	394
144	50	6	44	380	55	325
145	95	5	90	488	27	461
146	32	13	19	280	111	169
147	62	6	56	380	34	346
148	33	12	21	380	228	152
149	26	5	21	449	57	392
150	51	4	47	371	54	317
Total	2704	639	2065	18901	3484	15417
Average	54.08	12.78	41.3	378.02	69.68	308.34

6.4 Experimental Design

To demonstrate the validity of each of the evaluation hypotheses, a series of experiments have been conducted using an IF system-based methodology as in the standard TREC Filtering Track [Lewis et al., 2004, Robertson and Soboroff, 2002]. These extensive experiments were carried on the RCV1 50 assessors collections and their TREC relevance judgements. As mentioned previously, an IF system filters out irrelevant documents from a stream of incoming documents based on the user information needs. Out of different types of IF systems, including, batch, routing and adaptive IF systems, the routing system is adopted in the evaluation experiments mainly to avoid the tuning of any required thresholds and to test the performance of the system based on a ranked list of documents [Soboroff and Robertson, 2003].

Figure 6.8 illustrates the evaluation procedure implemented in this research. For each collection, the proposed models and frameworks are trained on the training set of the collection. A set of discovered relevant features (i.e., relevant features and their estimated weights learned from the training set) are used as a query $q = (t_1, t_2, t_3, \ldots, t_k)$ submitted to the IF system in which $q \subseteq T$ and $1 \leq k \leq |T|$. As in the TREC Filtering Track [Lewis et al., 2004, Robertson and Soboroff, 2002] and for each new document comes from the testing set, the system has to decide whether the new document is relevant to the user information needs, which are represented by the selected set of features (i.e., the query q in this case). A similar approach is applied to the baseline models.

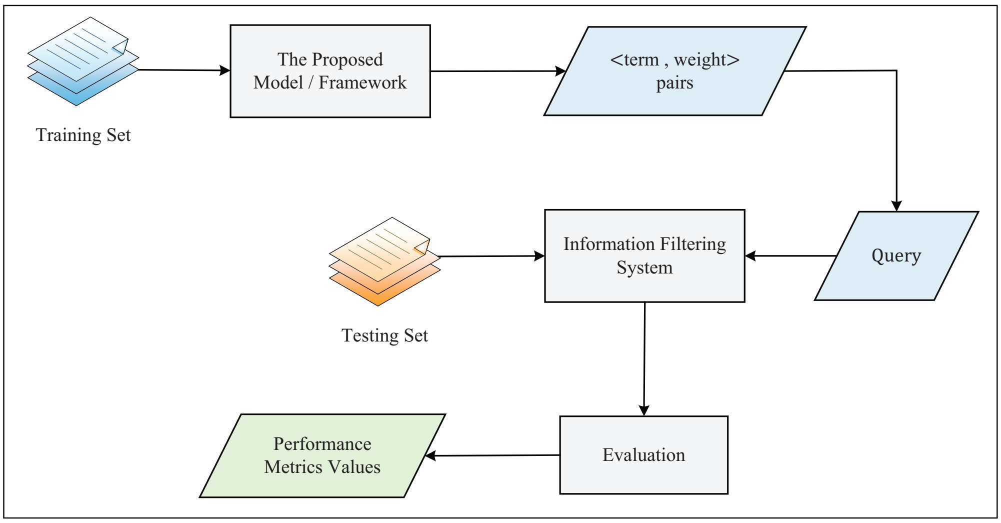

Training Set
The Proposed Model / Framework
<term , weight> pairs
Query
Information Filtering System
Testing Set
Evaluation
Performance Metrics Values

Figure 6.8: The main IF-based evaluation procedure.

Moreover, and in separate experiments, we used the terms of the TREC topics for the RCV1 dataset (see Appendix F) as relevant features. These terms are identified by the domain experts at NIST and will be used to evaluate the proposed TFS techniques and the baseline models in automatically discovering and ranking these features. Figure 6.9 illustrates this evaluation process. However, we do not assume that these terms are the only relevant ones in the used 50 collections, but to avoid the expense of manually identifying more relevant terms from these vast collections, we limited our study only to those terms presented in Appendix F.

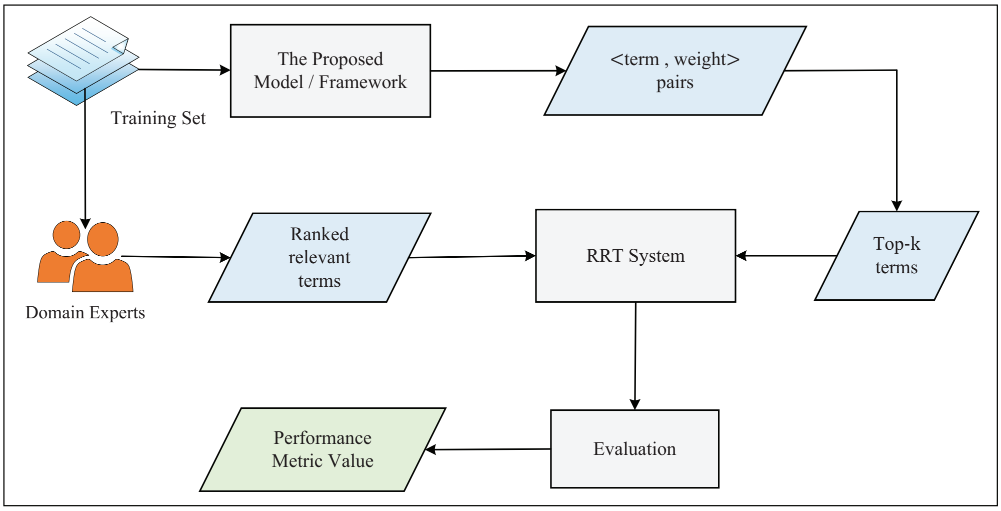


Figure 6.9: The RRT-based evaluation procedure.

If the results for the proposed models or frameworks are significantly better than the used baselines, then, it is valid to claim that the proposed technique reflects the developed hypothesis.

6.4.1 Unsupervised Learning Setting

Figure 6.8 briefly shows both the training and the testing stages of the evaluation process. Figure 6.10 further illustrates the training procedure of the proposed unsupervised models (i.e., SIF, SIF2 and UR) including the USIF framework. These models and the framework use only the relevant documents set D^+ in each collection as a domain-specific set of long documents. After completing the preprocessing steps on each set as previously shown in Figure 6.7, all documents paragraphs are split, stored in separate files and indexed for efficient mapping between a document and its paragraphs. Then, the LDA is used to extract some latent topics from all paragraphs in the collection. These topics are then used by the proposed SIF, SIF2 and the UR models in their fusion modules.

The solid arrows in Figure 6.10 show the sequential flow of these steps for the SIF, SIF2 and UR models while the dotted arrows display the subsequent training steps for the USIF framework. As the USIF framework utilises a document clustering algorithm in one of its

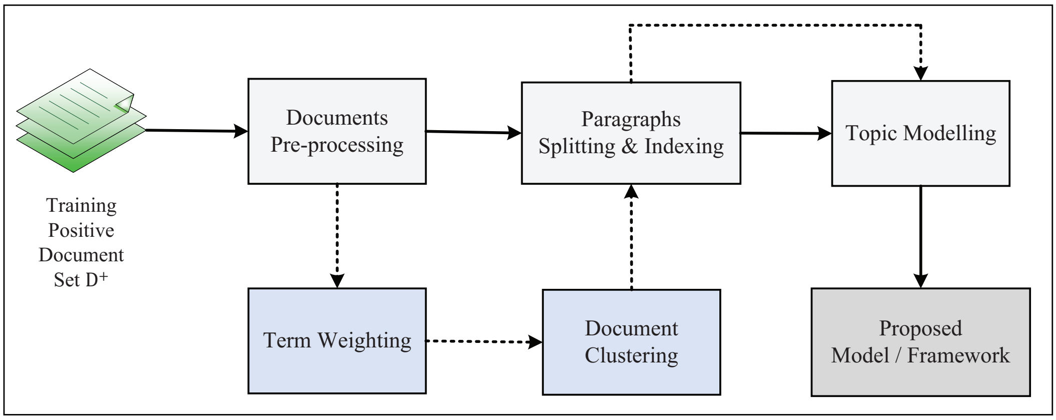


Figure 6.10: The training procedure for the proposed SIF, SIF2, UR models and the USIF framework.

stages, a term weighting scheme (i.e., TFIDF) is used on the preprocessed documents. Then, the bisecting k-means (BKM) algorithm [Savaresi and Boley, 2001] is used to cluster these documents based on the cosine similarity measure. Then, for each cluster formed by the BKM algorithm, the paragraphs of the documents in the cluster are split, and some latent topics are extracted using the LDA in a similar fashion as in training steps of the SIF, SIF2 and the UR models. Lastly, all unsupervised baseline models were trained as described in their original studies.

6.4.2 Supervised Learning Setting

In the training phase of the proposed SSIF framework, both relevant (positive) D^+ and irrelevant (negative) D^- training documents sets were used as shown in Figure 6.11. Each set is used separately, and the latent topics only extracted from D^+ because it is domain-specific and its subjects are more related to each other unlike the irrelevant set, which has a diverse collection of unrelated subjects. The negative set, on the other hand, is only used for the supervised learning algorithm, the support vector machine (SVM) in this research, which also requires the positive set as well. To train the SVM, a supervised term weighting scheme is used (i.e., BM25) to assign weights to terms in both documents sets after the preprocessing steps are completed. These weighted terms are used to represent all training documents (positive and negative) for the SVM. The SVM learns a hyperplane from these training documents, which can be used to separate between positive and negative information in those documents. The same training steps of the SIF, SIF2 and UR models are also applied on the relevant documents set for the SSIF framework. The solid green arrows in Figure 6.11 show the usage flow of the relevant documents while the red dotted arrows display the flow of negative documents. The generated latent topics of the LDA and the learned hyperplane of the SVM will be used by the fusion modules of the SSIF framework. Lastly, all supervised baseline models were trained as described in their original studies.

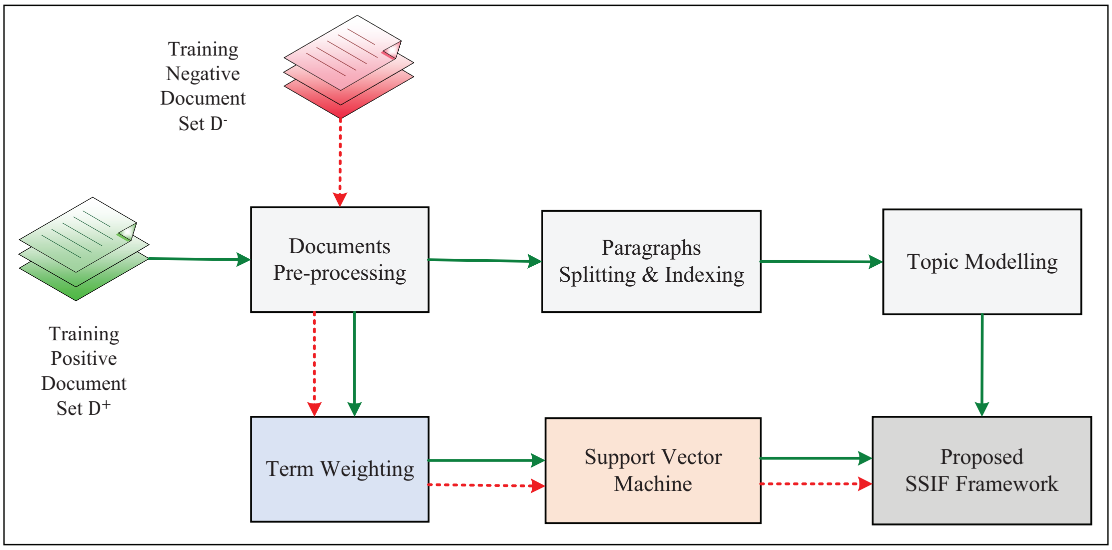


Figure 6.11: The training procedure for the proposed SSIF framework.

6.5 Baseline Models

For a more comprehensive evaluation, the performance of the proposed models and frameworks were compared to a wide range of TFS models. Over 20 different baseline models used for relevance discovery were selected and tested for IF and RRT tasks. These models use different types of text features, and they can be either supervised or unsupervised learning techniques. However, in this thesis, all the baseline models were categorised based on the feature fusion strategy they adopt. In the following sections, a short description is given for each model under its category, and more details about these baselines can be found in Chapter 2.

6.5.1 Early Fusion Models

Early fusion TFS models use low-level terms and consider no semantic information as described in Chapter 3. The following popular examples are selected as baselines in our evaluation experiments.

- **TFIDF** [Salton and Buckley, 1988]: is a widely accepted term weighting scheme in many IR applications. In an unsupervised manner, TFIDF assigns higher weights to terms that occur more frequently in a specific document.

- **Okapi BM25** [Robertson and Zaragoza, 2009]: is a popular, supervised document ranking algorithm in IR. It is term-based and its experimental parameters were set to $b = 0.75$ and $k_1 = 1.2$ in this thesis as recommended in [Gao et al., 2015, Manning et al., 2008b, Zhong et al., 2012].

- **Prob** [Jones et al., 2000a,b]: is a supervised probabilistic method that estimates the relevance weight of terms at the collection level.

- **Chi-square** (χ^2) [Chen and Chen, 2011]: is a popular statistical method that measures the informativeness of a term to its class information. It shows effective performance in supervised text classification compared to many other TFS [Tang et al., 2016, Yang and Pedersen, 1997].

- **MI** [Manning et al., 2008b]: mutual information is another supervised TFS that measures the mutual dependence between random terms and their classes information.

- **SVM** [Joachims, 2002]: support vector machine is a well-known supervised learning algorithm that discriminatively separates two different classes. Since IF can be considered another type of binary classification problem, the rank-based SVM was used in this research similarly as in [Algarni and Li, 2013, Gao et al., 2015, Li et al., 2008, Zhong et al., 2012].

- **LASSO** [Tibshirani, 1996]: the least absolute shrinkage and selection operator, commonly known as Lasso, is a supervised linear regression model. It can be used in TFS for relevance discovery as in [Li et al., 2015].

- **Rocchio** [Rocchio, 1971]: is widely used in IR, IF and text classification as a centroid-based classifier. It revises relevant terms weights using the negative training document set. In this study, Rocchio is used as in [Li et al., 2015, Wu et al., 2006].

6.5.2 Late Fusion Models

High-level features like phrases, patterns, topics, ontological concepts or a different combination of them contain different semantic information that makes them suitable for late feature fusion. A wide variety of popular and state-of-the-art late fusion models are selected as baselines in our experiments. They are briefly described as follows:

- n**-grams**: is a standard phrase-based model that uses n-grams extracted from relevant documents to represent user information needs, where, as in [Albathan et al., 2012, 2014, Gao et al., 2015], the best value of n is empirically set to 3 (a tri-gram).

- **PCM** [Albathan et al., 2012]: is the pattern co-occurrence matrix model that removes noisy patterns extracted from a set of relevant documents paragraphs. PCM is unsupervised and utilises a pattern co-occurrence matrix to identify interesting set of closed sequential patterns for relevance discovery.

- **SCSP** [Albathan et al., 2014]: is a supervised pattern-based TFS model. It extends a random-set to find specifically relevant closed sequential patterns extracted from both positive and negative documents.

- **PLSA** [Hofmann, 2001]: is an unsupervised topic-based TFS model. It identifies hidden topics from a set of documents that can be used to represent user information needs. These topics can alleviate the problem of polysemy to a certain extent as mentioned previously. PLSA is a probabilistic enhancement to the latent semantic analysis (LSA) model [Deerwester et al., 1990].

- **LDA** [Blei et al., 2003]: is the most widely used topic modelling algorithm. Unlike the PLSA, LDA is theoretically sound and more effective. It probabilistically generates latent topics from a collection of documents in an unsupervised way. In our experiments, PLSA and LdaDoc were trained on D^+ documents while LdaPara was trained on D^+ paragraphs.

- **TNG** [Wang et al., 2007]: is a topical n-grams TFS model that integrates topic modelling with phrases to discover topical phrases that are more discriminative and interpretable. TNG is treated as a relevance ranking model in our experiments as in [Gao et al., 2014b, 2015].

- **PBTM-FP** [Gao et al., 2013]: is an unsupervised TFS model that incorporates latent topics and frequent patterns (FP) to obtain a more semantically rich and discriminative representation to be used for IF.

- **PBTM-FCP** [Gao et al., 2013]: is similar to the PBTM-FP model except it uses the frequent closed pattern (FCP) instead in order to reduce the size of FP as redundant patterns .

- **SPBTM** [Gao et al., 2014b]: is a third extension to PBTM-FP and PBTM-FCP. It uses significant matched patterns (i.e., significantly representative and specific frequent patterns) to represent latent topics.

- **MPBTM** [Gao et al., 2015]: is a more advanced extension to the PBTM-FP, PBTM-FCP and SPBTM models. It uses maximum matched patterns (i.e., the most representative and specific frequent patterns) to represent latent topics.

- **LdaConcept** [Chemudugunta et al., 2008]: combines topic models with ontological concepts to semantically represent user information needs. LdaConcept is unsupervised and is similarly adopted in our experiments as in [Bashar and Li, 2017, Bashar et al., 2016].

6.5.3 Hybrid Fusion Models

A hybrid fusion can be developed through the combining of early and late fusions strategies to exploit the advantages of both low- and high-level text features in a unified framework. Three state-of-the-art baseline models are selected to represent this type of fusion.

- **PDS** [Zhong et al., 2012]: is a pattern deploying technique based on support. It is an unsupervised extension to the PTM model [Wu et al., 2006]. PDS uses high-level patterns extracted from relevant documents to accurately weight low-level terms to represent what the user wants.

- **MP** [Yan et al., 2005]: similar to the PDS model, the master pattern technique uses patterns to identify relevant low-level terms. Instead of deploying patterns, MP summarises or groups text patterns into L clusters (aka pattern profiles) based on defined similarities. MP is used in our experiments as in [Bashar and Li, 2018, Bashar et al., 2016, 2017].

- **RFD$_2$** [Li et al., 2015]: is a supervised relevant feature discovery model. It is an extension of RFD$_1$ [Li et al., 2010] and uses high-level patterns to discover relevant low-level terms that are clustered into three distinct groups; positive specific, general and negative specific sets of terms. RFD$_2$ experimental parameters are kept in our experiments as the original study.

6.6 Performance Measures

Measuring the performance of an information system is an essential step in any experimental evaluation process. In our experiments, the effectiveness of the proposed TFS models and frameworks in IF is measured by six different evaluation metrics that are well-established and commonly used in the IR and IF research communities. These measures are derived, in a way or another, from the standard effectiveness metrics; precision and recall. The six metrics are the average precision of the top-20 ranked documents (P@20), break-even point (BP), mean average precision (MAP), F measure, interpolated average precision (IAP) and the interpolated precision averages at 11 standard recall levels (11-point). Each of these measures concerns about a particular aspect of the model overall performance as it will be described in this section. More information about these measures can be found in [Manning et al., 2008a].

The previous six metrics are used to measure the effectiveness of our proposed techniques for IF specifically in returning relevant documents from the testing sets of the first 50 collections of the RCV1 dataset. However, there is a need to measure the effectiveness of the proposed techniques in identifying and ranking relevant features (i.e., terms) that are identified and selected by TREC's domain experts. The standard, normalized discounted cumulative gain (nDCG) measure is used for this task and it will be described below in this section. Moreover, in our experiments, the percentage change and the Student's t-test are used to analyse the significance of the difference between the results of the proposed models and frameworks and the selected baselines.

6.6.1 Precision and Recall

For an IR system, the precision is the ”fraction of retrieved documents that are relevant” to the user query while the recall is the ”fraction of relevant documents that are retrieved” [Manning et al., 2008a]. For a text classifier, the accuracy measure, which is the fraction of the classifier's predictions that are correct, is usually used instead of the precision and recall. The confusion matrix depicted in Table 6.3, which is a special type of contingency table, is used for binary classification judgement. Therefore and based on the accuracy definition, a classifier's accuracy $= (\text{TP} + \text{TN}) \div (\text{TP} + \text{FP} + \text{FN} + \text{TN})$, where TP is the number of documents that the classifier identified as relevant, TN is the number of documents that the classifier identified as irrelevant, FP is the number of documents that the classifier incorrectly identified as relevant and FN is the number of relevant documents that the classifier could not identify [Manning et al., 2008a, Wu, 2007].

Since an IR or IF system can be considered as a two-class classifier (i.e., relevant-irrelevant), it implies that the accuracy measure can be used in measuring IR and IF systems performance.

Table 6.3: The confusion matrix of classification

		Human judgement	
		Yes	No
System judgement	Yes	**True Positive** (TP)	Fale Positive (FP)
	No	False Negative (FN)	**True Negative** (TN)

However, accuracy is not suitable for retrieval and filtering problems as it is biased toward the bigger class in the used dataset [Manning et al., 2008a]. For example, if an imbalanced dataset has 98.9$\%$ positive samples, then, a classifier can achieve 98.9$\%$ accuracy by just classifying all documents as positive and vice-versa when most samples are negative. Instead, the precision and recall are more suitable for IR and IF because users are only interested in positive class [Algarni, 2011]. Therefore and in a similar context as in the accuracy measure above, Table 6.3 can be used to calculate precision (P) and recall (R) as $\mathrm{P} = \mathrm{TP} \div (\mathrm{TP} + \mathrm{FP})$ and $\mathrm{R} = \mathrm{TP} \div (\mathrm{TP} + \mathrm{FN})$, respectively.

6.6.2 Effectiveness Measures

Based on the definitions of precision and recall, more practical metrics can be derived to solve some problems that precision and recall alone cannot resolve.

- **Break-even Point** (BP): Break-even point is a commonly used measure in the area of IR and IF. It concerns about the relationship between P and R and indicates the point when both P and R values are equal ($\mathrm{P} = \mathrm{R}$). Thus, the higher the value of the BP measure, the more effective the evaluated system is.

- **F measure** ($\mathrm{F}_{\beta=1}$): F measure is another metric that concerns about the relationship between P and R. Unlike the BP metric, which only concern about $\mathrm{P} = \mathrm{R}$, F measure can be used to trade off between P and R because usually, in a testing set, R can be high and P may be low and vice versa. F measure is the weighted harmonic mean of P and R and can be calculated as F measure $= ((1 + \beta^2) \times \mathrm{P} \times \mathrm{R}) \div (\beta^2 \times \mathrm{P} + \mathrm{R})$. As the harmonic mean tends to be closer to the smallest value of either P or R, F measure is used in our experiments when the value of both R and P wanted to be high rather one higher, and the other is lower and vice versa. Thus, we set the parameter β to be equal to 1 ($\beta = 1$) which simplifies the last equation to $\mathrm{F}_{\beta=1} = (2 \times \mathrm{P} \times \mathrm{R}) \div (\mathrm{P} + \mathrm{R})$.

Despite the practicality of the P, R, BP and $\mathrm{F}_{\beta=1}$ measures, they ignore the order of the retrieved documents [Manning et al., 2008a]. It is assumed that in a ranked list of documents the top u documents are more relevant than those at the end of the list. Also, $\mathrm{F}_{\beta=1}$ and BP, as single-valued metrics, do not provide a more detailed picture of the whole system performance. To further address these issues, four effective measures are adopted in our experiments and described as follows:

- **Precision at top-u documents** (P@u): Precision takes into account all retrieved documents by the IR/IF system, but a user might be interested only in the first or two dozens of documents (i.e., a specified cut-off) ordered based on their relevance to user information needs. Thus, in this research, the precision of the top u returned documents (P@u) is used, and the value of u is set to be 20 in our experiments, which is an agreeable number within IR and IF communities [Gao et al., 2015, Manning et al., 2008a, Zhong et al., 2012].

- **Mean Average Precision** (MAP): It is the most commonly used single-figure metric among the TREC community [Manning et al., 2008a]. MAP can be calculated by first measuring P at each relevant document in a ranked list of retrieved documents based on their relevance to a user information need (aka topic or collection), and, then averaging P over all topics (collections) in the testing sets. MAP provides an excellent indication about the quality of the evaluated system as it combines the measurements of P, overall R and the relevance ranking of the retrieved documents.

- **Interpolated precision averages at 11 standard recall levels** (11-point): It is an effective measure for comparing the performance of two or more different IR/IF systems in distinctive details. 11-point metric is the interpolated P at 11 standard R-levels. This measure examines the entire P-R curve at only 11 points $(0.0, 0.1, 0.2, ..., 1.0)$ where the first R point is equal to zero, which is the smallest value (e.g., $1 \div (\mathrm{TP} + \mathrm{FN})$) [Algarni, 2011].

- **Interpolated Average Precision** (IAP): Unlike the 11-point measure, the IAP is a single-valued metric that can be calculated by averaging the interpolated P at 11 standard R-levels for one topic (i.e., TREC topic), in a similar fashion as in MAP, and, then averaging for all topics.

As mentioned previously, the IF system in our experiments returns a ranked list of documents after accepting a query q, which is a sequence of $\langle \text{term}, \text{weight} \rangle$ pairs discovered by the proposed TFS techniques and the baseline models. The query q in this research represents a user information need. All the six metrics discussed above are used to measure the effectiveness of the IF system in identifying relevant documents from the testing sets of the RCV1 dataset based on q. However, there is a need to measure the performance of our proposed techniques as well as the used baseline models in RRT. The nDCG measure at top-k terms is used for this task as described below.

- **normalized Discounted Cumulative Gain at top-k feature** (nDCG@k): It is commonly used within the IR/IF community to measure the effectiveness of IR/IF models in ranking highly relevant documents. nDCG is sensitive to the position of the relevant document, so as it rewards highly ranked documents, it also penalises those in lower ranks. Further details about the nDCG metric can be found in [Järvelin and Kekäläinen, 2002]. However, we adopted the nDCG to measure the effectiveness of the proposed models and frameworks and the baseline models in ranking relevant terms instead of documents. We used the terms of each TREC topic as our relevance judgment as described previously. As nDCG is usually used for graded relevance judgment, it can also be used for binary one as in our case. The nDCG at top-k terms is calculated in our experiments based on its formula in [Manning et al., 2008a].

6.6.3 Statistical Significance Measures

It is a common practice in scientific research to analyse experimental results using some well-established mathematical tools. Two statistical significance measures are used to evaluate the reliability and significance of the results of our experiments. These measures are the Percentage of Change and the Student's Paired T-Test [Smucker et al., 2007, Urbano et al., 2013]. They are described as follows:

- **Percentage Change** (improvement$\%$): It is commonly used to calculate the difference between two mean values and show how statistically significant this difference is in a percentage format [Gao et al., 2015, Li et al., 2015, 2010, Wu et al., 2006]. In our experiments, the *"improvement$\%$"* is used to denote the result of this test in several tables. The percentage change between two TFS models can be calculated as $\text{improvement}\% = (\nu_{\text{our}} - \nu_{\text{base}}) \div \mid \nu_{\text{base}} \mid$, where ν refers to the result of an experimental model (in our case, the result is averaged over the used 50 collections) and $\mid \nu \mid$ is the absolute value of that result.

- **Student's Paired T-Test** (t-test): It is also widely used in IR and IF to statistically measure how significant the improvement is between two related sets of results [Smucker et al., 2007, Urbano et al., 2013]. The t-test assesses the mean of two different numerical groups and shows how significant is the difference between their values. Usually, the null hypothesis of this test assumes that no significant difference exists between the results. However, this hypothesis can be either rejected or accepted based on the *p-value* of the test. If the *p-value* is less than 0.05 $(\text{p-value} < 0.05)$, it indicates that the difference between these two groups is significant, and the null hypothesis can be rejected and vice versa.

6.7 Experimental Settings

All experiments described in this chapter were conducted on a personal computer (PC) equipped with an Intel® Core™ i7-4510U @ 2.00 GHz processor and a main memory of 8.00 GB running on Microsoft® Windows® 10 Pro. The proposed models and frameworks and the IF evaluation system including all baseline models were implemented in the Java 8 programming language (JRE 8.0.31) using the NetBeans IDE (version 8.0.2). The RCV1 dataset was obtained from a TREC licensed CD, and its relevance judgement information was downloaded from the TREC website. [2]

The MALLET toolkit [McCallum, 2002] [3] was used to implement all LDA-based models and frameworks except for the PLSA model where the Lemur toolkit [4] was used instead. All topic-based models require some parameters to be set. For the LDA-based models, the number of iterations for the Gibbs sampling was set to be 1000 and for the hyper-parameters to be $\alpha = 50/V$ and $\beta = 0.01$ as they were justified in Steyvers and Griffiths [2007]. The number of iterations for the PLSA was configured to be 1000 (default setting). Lastly, it should be mentioned that the LDA training needs only to be done once and off-line.

The CLUTO toolkit [5] was used to cluster the relevant documents of each collection into hard clusters using its graphical tool gCLUTO [Rasmussen and Karypis, 2004]. The repeated

[2] https://trec.nist.gov/data/t2002_filtering.html
[3] http://mallet.cs.umass.edu
[4] https://www.lemurproject.org/
[5] http://glaros.dtc.umn.edu/gkhome/views/cluto

bisecting algorithm is selected to be the clustering technique used with the USIF framework. Other parameter settings in the gCLUTO environment are set as follows: the similarity function is set to be the cosine; I2 is the criterion function; 10 for the number of trials; the default values are accepted for the remaining parameters.

It is challenging to predetermine the optimal number of clusters for a given collection [Das et al., 2008, Jain, 2010, Liu and Croft, 2004]. However, in this research, and based on the USIF performance on a sample of collections, the straight line equation $L = mX + b$ was fitted through the number of clusters L and the number of relevant documents X, where m is the slope and b is the bias. They were empirically set to be $m = 0.5$ and $b = 0.5$.

6.8 Results

In this section, the experimental results of the proposed TFS models and frameworks are presented and compared with the results of various baseline techniques. The results show the effectiveness of the proposed fusion-based techniques in IF using the 50 domain experts assessed collections of the RCV1 dataset. The effectiveness is measured by six standard evaluation metrics. The results also demonstrate the effectiveness of our proposed models and frameworks in identifying and ranking relevant features that describe user information needs. The standard nDCG measure is used for evaluating the quality of these identified features based on the relevance judgment of NIST domain experts. Additionally, two statistical significance tests, namely the percentage change and the t-test, are used to detect and verify the improvement in each result compared to the baselines.

These experimental results are presented in the following sections: the result of the SIF model compared to the baseline models are given in Section 6.8.1. The results of the SIF2 model and the comparisons with the baselines are given in Section 6.8.2. The UR method improvements to many existing relevance discovery models are demonstrated by the results presented in Section 6.8.3. Section 6.8.4 shows the results of the USIF framework compared to the used baseline models and Section 6.8.5 presents the results of the SSIF framework in a similar way.

6.8.1 The Proposed SIF Model

The results of the SIF model and the selected baselines are illustrated in Table 6.4 and Figure 6.12 (left). These experimental results show the effectiveness of SIF and the baseline models for the IF task measured by the standard metrics of P@20, BP, MAP, $F_{\beta=1}$, IAP and 11-point. The baseline models in Table 6.4 are categorised based on the type of text feature they use. Table 6.5 and Figure 6.12 (right) illustrate the results of the SIF model and some baseline models for RRT measured by the nDCG metric. These results, in Tables 6.4 and 6.5, are the average of the 50 collections of the RCV1.

Moreover, Table 6.6 and the p-value column in Table 6.5 illustrate the results of the statistical significance measure, the t-test, and the ”improvement%”, in Tables 6.4 and 6.5, represents the percentage change, in our SIF model's performance compared to the best result of the baseline model. We consider any improvement in the percentage change that is greater than 5.0% to be significant. From all these tables and figures, we can see that the SIF model outperformed all baseline methods in all measures for all experimental tasks. More evaluation details are given in the following sections.

- **SIF Versus Term-based Models**

  The BM25 and the TFIDF models were selected to represent the term-based category and their experimental performance in IF and RRT tasks were compared to the proposed SIF model. While BM25 maintained superior performance in both experimental tasks compared to the TFIDF as can be seen in Table 6.4 and Figure 6.12 (left) and in Table 6.5 and Figure 6.12 (right), respectively, our SIF model outperformed BM25 for IF in all five measures by an overall average improvement of 20.866% with a minimum of 14.237% on $F_{\beta=1}$ and a maximum of 27.416% on P@20. Figure 6.12 (left) clearly shows the superior performance of the SIF model in IF compared to the BM25 measured by the 11-point metric.

  Also, our model significantly outperformed BM25 in RRT by an average improvement of 448.159% measured by the nDCG metric using just the top-4 keywords (i.e., k = 4) ranked by each model as illustrated in Table 6.5. While k = 4 is the average number of terms in the 50 titles of TREC topics (see Appendix F), Figure 6.12 (right) shows that our SIF model was consistently significant in RRT compared to both BM25 and TFIDF at all top-k values. The percentage change test results in Tables 6.4 and 6.5 show that all the performance improvements of the SIF model in IF and RRT over the BM25 were statistically very significant as they were much higher than 5.0%. The t-test results in Table 6.6 and 6.5 confirmed this significance because the p-values were much less than 0.05 in both tails of the test.

- **SIF Versus Phrase-based Models**

Table 6.4: SIF results for the IF task compared to the baselines (grouped based on the type of feature used by the model) for all measures averaged over the first 50 collections of the RCV1 dataset

Model	P@20	BP	MAP	F$_{\beta=1}$	IAP
SIF	**0.567**	**0.475**	**0.500**	**0.473**	**0.527**
LDA	0.492	0.414	0.442	0.437	0.468
PLSA	0.423	0.386	0.379	0.392	0.404
improvement%	+**15.337**%	+**14.773**%	+**13.273**%	+**8.141**%	+**12.507**%
PDS	0.496	0.430	0.444	0.439	0.464
MP	0.426	0.392	0.393	0.409	0.421
improvement%	+**14.315**%	+**10.388**%	+**12.805**%	+**7.687**%	+**13.524**%
n-grams	0.401	0.342	0.361	0.386	0.384
improvement%	+**41.397**%	+**38.936**%	+**38.608**%	+**22.526**%	+**37.287**%
BM25	0.445	0.407	0.407	0.414	0.428
TFIDF	0.354	0.338	0.337	0.367	0.366
improvement%	+**27.416**%	+**16.620**%	+**22.981**%	+**14.237**%	+**23.076**%
PBTM-FCP	0.489	0.420	0.423	0.422	0.447
PBTM-FP	0.470	0.402	0.427	0.423	0.449
TNG	0.447	0.360	0.372	0.386	0.394
improvement%	+**15.951**%	+**13.087**%	+**17.214**%	+**11.856**%	+**17.220**%

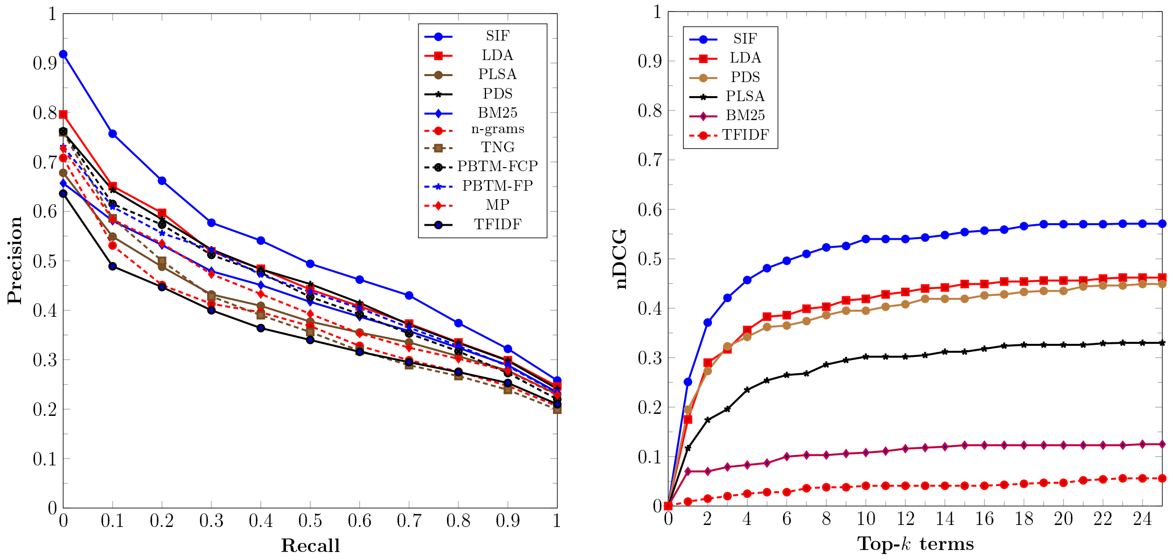


Figure 6.12: The 11-point results for IF (left) and the nDCG@k results for RRT (right) of SIF in comparison with baselines averaged over the first 50 collections of the RCV1 dataset.

Table 6.5: The SIF results for the RRT task including the percentage change and the t-test p-value in comparison with some of the baselines averaged over the first 50 collections of the RCV1 dataset

Model	nDCG@4	improvement%	p-value
SIF	**0.457**	**0**%	N/A
LDA	0.356	+**28.132**%	6.581E-04
PDS	0.342	+**33.536**%	3.504E-04
PLSA	0.235	+**94.085**%	3.055E-05
BM25	0.083	+**448.159**%	8.041E-11
TFIDF	0.025	+**1706.215**%	8.939E-12

Table 6.6: The t-test p-values of the best baseline model in each category in comparison with the SIF model for the IF task results in Table 6.4

Model	Tail(s)	P@20	BP	MAP	$F_{\beta=1}$	IAP
LDA	One	7.557E-04	5.117E-07	4.785E-05	7.002E-05	1.239E-05
	Two	1.511E-03	1.023E-06	9.571E-05	1.400E-04	2.477E-05
PDS	One	3.435E-03	3.969E-03	9.530E-04	2.726E-03	1.298E-04
	Two	6.869E-03	7.937E-03	1.906E-03	5.451E-03	2.596E-04
n-grams	One	4.091E-09	3.483E-11	1.280E-12	5.943E-11	2.051E-13
	Two	8.181E-09	6.967E-11	2.560E-12	1.189E-10	4.102E-13
BM25	One	1.440E-04	1.103E-03	8.550E-05	1.065E-04	1.110E-05
	Two	2.879E-04	2.206E-03	1.710E-04	2.129E-04	2.220E-05
PBTM-FCP	One	8.335E-03	3.411E-03	1.444E-04	9.346E-04	6.542E-05
	Two	1.667E-02	6.823E-03	2.889E-04	1.869E-03	1.308E-04
PBTM-FP	One	5.664E-04	4.085E-05	1.271E-04	1.342E-04	1.965E-05
	Two	1.133E-03	8.171E-05	2.541E-04	2.683E-04	3.929E-05

For the phrase-based category, the n-grams language model was used as a baseline for the IF only. It was not used for the RRT task as it does not explicitly weight single terms. As shown in Table 6.4, the SIF model significantly outperformed the n-grams model in all measures by an overall average improvement of 35.751$\%$ with a minimum of 22.526$\%$ on the F$_{\beta=1}$ metric and a maximum of 41.397$\%$ on the P@20 measure. Moreover, the 11-point result in Figure 6.12 (right) illustrates the superiority of SIF over the n-grams model and confirms the significant improvements that were shown in Table 6.4. All SIF improvements were much higher than 5.0$\%$, and its t-test p-values in Table 6.6 were largely less than 0.05, indicating that SIF improvements were statistically very significant.

- **SIF Versus Pattern-based Models**

  Our SIF model continues to perform significantly better than the state-of-the-art pattern-based techniques represented in our experiments by the PDS and the MP models. For the IF and the RRT tasks, SIF results were compared to the PDS because it scored better results than the MP model as illustrated in Table 6.4, and can rank relevant terms while the MP does not deal with individual terms.

  In Table 6.4, SIF outperformed the PDS in all measures on average by a minimum improvement of 7.687$\%$ and a maximum of 14.315$\%$ on the F$_{\beta=1}$ and the P@20 respectively. Our SIF model maintained an average improvement of 11.744$\%$ over the PDS. Also, Figure 6.12 (left) illustrates the superiority of SIF compared to PDS on the 11-point measure. For the RRT task, SIF was significantly better than the PDS by 33.536$\%$ on the nDCG@4 as shown in Table 6.5 and continues to perform consistently better with different k values as illustrated in Figure 6.12 (right). All SIF improvements over the PDS were statistically significant as confirmed by the percentage change as well as the t-test results in Tables 6.4, 6.5 and 6.6, respectively.

- **SIF Versus Topic-based Models**

  We selected LDA and its predecessor, the PLSA, as baseline models to represent this category. LDA continues to achieve better results for the IF and RRT tasks than the PLSA. Therefore, our SIF model will be compared to LDA rather than the PLSA. As illustrated in Table 6.4, the SIF model outperformed the LDA for IF in all measures. Our model achieved a minimum improvement of 8.141$\%$ on the F$_{\beta=1}$ measure over the LDA, and a maximum improvement of 15.337$\%$ on the P@20 over the same model. On an overall average, the SIF model scored an improvement of 12.806$\%$ over the LDA in the IF task. This improvement can be clearly seen in Figure 6.12 (left) on the 11-point measure.

For the RRT task measured by the nDCG metric, SIF continues to outperform the LDA by an average improvement of 28.132% as illustrated in Table 6.5 using only the top 4 keywords from each training collection of the RCV1 dataset. Moreover, our model seems to be insensitive to the value of the k parameter for the RRT task as can be seen in Figure 6.12 (right). SIF consistently performed very significantly on all k values compared to the LDA model. The percentage change in Tables 6.4 and 6.5 represented by the ”improvement%” shows that all SIF improvements are statistically significance because they are all over 5.0%. The t-test results in Tables 6.6 and 6.5 confirmed these statistical significances as all p-values were much less than 0.05 in all tails of the test.

- **SIF Versus Hybrid Features-based Models**

  Three models were selected for this category. The pattern-based topic models (i.e., PBTM-FP and PBTM-FCP) performed better than the topical N-grams (TNG) model in the IF task. As all these models were not developed to deal with individual terms explicitly, they were not used for the RRT task. For the IF task and according to Table 6.4, the SIF model outperformed both PBTM models in all measures. SIF scored a minimum improvement of 11.856% over the PBTM-FP model on the $F_{\beta=1}$ measure and a maximum improvement of 17.220% on the IAP metric over the same baseline model. Overall, our model maintained an overage improvement of 15.066% in all metrics over the two PBTM models. The 11-point result in Figure 6.12 (left) confirmed this improvement over all baseline models, including the PBTMs. Moreover, the percentage change and the t-test results in Tables 6.4 and 6.6, respectively, show that SIF improvements over the baselines were statistically significant as these improvements were higher than 5.0% and their one- and -two-tailed p-values were less than 0.05.

Based on the results presented earlier, we are confident in claiming that our SIF model can effectively generalise the local term weight at the document level in the LDA term weighting function and, thus, provide a more globally representative weight when it combined with the term document frequency. Also, SIF is more effective in selecting relevant features to acquire user information needs that represented by a set of long documents. Overall, these results support the hypothesis 1.

Despite its effectiveness, SIF was built on the hypothesis of identical topical terms are equally important in all relevant documents. We argued that such an assumption could be too simple and need to be relaxed. Therefore, we revisited SIF and extended it to SIF2. The following section shows the experimental results of SIF2 for the same experimental tasks.

6.8.2 The Proposed SIF2 Model

This section presents the experimental results of the SIF2 model that has been introduced in Chapter 4. As in our SIF model, SIF2 was also tested for an IF application and its performance was measured by six different effectiveness metrics. SIF2 was also experimentally examined for the RRT task, and its performance was measured by the nDCG@k metric. Two groups of different baseline models, supervised and unsupervised, were used for comparison with our new model. These baseline models were examined for the same IF and RRT tasks. However, those baselines that do not have an explicit mechanism for ranking terms were exempted from the RRT task. The detailed comparisons are given below based on these two groups.

Table 6.7 and Figure 6.13 (left) illustrate SIF2 results as well as the baselines for the IF system while Table 6.8 and Figure 6.13 (right) show the results for the RRT task. The percentage change and the t-test results were presented in Tables 6.7, 6.8 and 6.9.

- **Comparisons with Supervised Models**

  The upper part of Table 6.7 summarises the results of SIF2 and three supervised baseline models for the IF task. These supervised models are the term-based SVM and BM25 and the pattern-based SCSP model. The results are sorted in descending order, and SIF2's results are only compared with the best baselines.

  As can be seen from Table 6.7, our model outperformed the SVM in all measures. It maintained an overall average improvement of 20.117$\%$ over the SVM with a minimum improvement of 12.357$\%$ and a maximum of 23.218$\%$ on the $\text{F}_{\beta=1}$ and P@20 measures, respectively. This significant improvement can be seen clearly using the 11-point result in Figure 6.13 (left) in which the SIF2 model outperformed all the baselined models in general and the supervised ones more specifically.

  For the RRT task results in Table 6.8, SIF2 also kept its superiority over the SVM with an average improvement of 713.793$\%$ using only four terms. While BM25 scored better than the SVM in this task, our model was significantly better than the BM25 by an average improvement of 466.775$\%$. Moreover, Figure 6.13 (right) shows the significant performance of SIF2 in the RRT experiment compared to all baseline models, including the supervised ones. The Figure also illustrates our model insensitivity to the hyperparameter k in which it scored much higher than any baseline model at any given k value.

According to the percentage change test, all SIF2 improvements over the supervised models presented in Tables 6.7 and 6.8 were statistically significant as they were all much higher than 5.0%. The t-test results in Tables 6.9 and 6.8 further confirmed the statistical significance of SIF2 results as each p-value of the test is much lower than 0.05 for all measures in the two tails of the t-test.

Table 6.7: The SIF2 results for the IF task compared to the baselines (grouped as supervised and unsupervised) for all measures averaged over the first 50 document collections of the RCV1 dataset

Model	P@20	BP	MAP	$F_{\beta=1}$	IAP
SIF2	**0.605**	**0.504**	**0.535**	**0.491**	**0.557**
SVM	0.491	0.414	0.436	0.437	0.462
SCSP	0.480	0.407	0.420	0.423	0.442
BM25	0.445	0.407	0.407	0.414	0.428
improvement%	+**23.218**%	+**21.739**%	+**22.706**%	+**12.357**%	+**20.563**%
SPBTM	0.527	0.448	0.456	0.445	0.478
PDS	0.496	0.430	0.444	0.439	0.464
LdaPara	0.492	0.414	0.442	0.437	0.468
PBTM-FP	0.470	0.402	0.427	0.423	0.449
PBTM-FCP	0.489	0.420	0.423	0.422	0.447
LdaDoc	0.457	0.391	0.400	0.413	0.434
PLSA	0.423	0.386	0.379	0.392	0.404
TNG	0.447	0.360	0.372	0.386	0.394
n-grams	0.401	0.342	0.361	0.386	0.384
improvement%	+**14.801**%	+**12.454**%	+**17.425**%	+**10.227**%	+**16.473**%

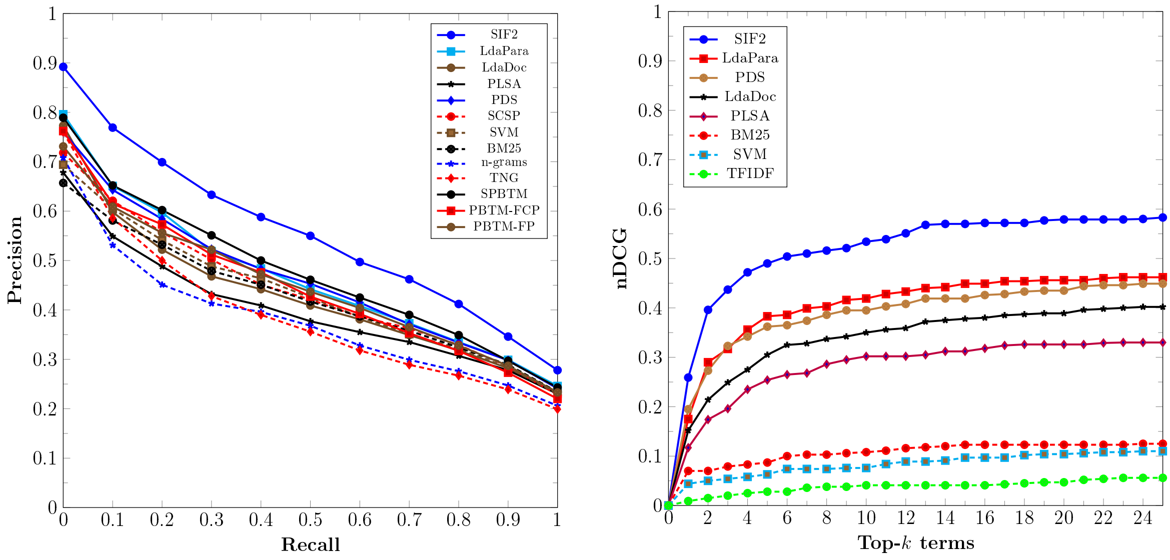


Figure 6.13: The 11-point results for IF (left) and the nDCG@k results for RRT (right) of SIF2 in comparison with baselines averaged over the first 50 collections of the RCV1 dataset.

Table 6.8: The SIF2 results for the RRT task including the percentage change and the t-test p-value in comparison with some of the baselines averaged over the first 50 collections of the RCV1 dataset

Model	nDCG@4	improvement%	p-value
SIF2	**0.472**	**0**%	N/A
LdaPara	0.356	+**32.483**%	8.057E-05
PDS	0.342	+**38.071**%	6.678E-05
LdaDoc	0.275	+**71.636**%	5.255E-06
PLSA	0.235	+**100.676**%	5.115E-06
BM25	0.083	+**466.775**%	6.530E-12
SVM	0.058	+**713.793**%	2.821E-13

Table 6.9: The t-test p-values of the best baseline model in each category in comparison with the SIF2 model for the IF task results in Table 6.7

Model	Tail(s)	P@20	BP	MAP	$F_{\beta=1}$	IAP
SPBTM	One	6.244E-03	1.294E-02	2.393E-04	1.048E-03	1.638E-04
	Two	1.249E-02	2.588E-02	4.785E-04	2.096E-03	3.276E-04
SVM	One	1.847E-04	1.970E-04	6.051E-06	2.346E-05	2.904E-06
	Two	3.694E-04	3.940E-04	1.210E-05	4.693E-05	5.809E-06

- **Comparisons with Unsupervised Models**

  In a similar setting, our SIF2 model was compared to a wide range of unsupervised baseline TFS models that use different text features. The SPBTM technique uses a combination of high-level features (i.e., patterns and latent topics) and achieved the best result among the other baseline models. Thus, SIF2 performance was compared to the SPBTM's for the IF task and to the LDA instead for the RRT as the SPBTM does not deal with low-level terms and LDA was the best unsupervised baseline model in this task.

  In IF, our model scored much higher than the SPBTM in all measures by an overall average of 14.276%, as shown in Table 6.7. SIF2's lowest average improvement was 10.227% on the $F_{\beta=1}$ metric and its highest average improvement was 17.425% on the MAP measure. Further, the superiority of our model over the SPBTM can be seen clearly on the 11-point measure, which is illustrated in Figure 6.13 (left). SIF2 scored much higher precision than the SPBTM model at every standard point of the 11 recall-level. In the RRT task, our SIF2 model outperformed the LDA by an average improvement of 32.483% using only four keywords as shown in Table 6.8. Figure 6.13 (right) clearly demonstrates the significant performance of the SIF2 model in the RRT experiment compared to the unsupervised baseline models not only using four terms but at all used terms (i.e., top-25 keywords).

As per the percentage change measure, all SIF2 improvements over the SPBTM and the LDA were statistically significant because they were much greater than 5.0% as shown in Tables 6.7 and 6.8. The two-tailed p-values of the t-test confirmed the statistical significance of the SIF2 results as illustrated in Tables 6.9 and 6.8.

As per the results reported earlier, we can claim in much confidence that our SIF2 model managed to relax the SIF hypothesis. It can effectively estimate a more accurate weighting function that measures the importance of topical terms in each relevant document. SIF2 also can better select relevant features from a document collection that discuss user information needs via the fusion of the estimate weighting function with a more representative global statistic. Therefore, the reported results support hypothesis 2.

SIF and SIF2 models managed to deal with some uncertainties when estimating the relevance of topical terms in a collection of relevant documents. In the following section, we experimentally demonstrate the effectiveness of our UR method in dealing with uncertainties in relevant features that are discovered by various TFS models and techniques.

6.8.3 The Proposed UR Method

In this section, the experimental evaluation of the proposed UR method is presented. The UR method has been introduced in Chapter 4 to deal with uncertainties in relevant features discovered by various supervised and unsupervised TFS models. Unlike the experiments of our SIF and SIF2 models, the UR method is integrated with each baseline model to scale and then re-rank its weighted relevant terms. The integration produces an improved baseline model, called 'iModel' (e.g., iSVM), which is experimentally examined for the IF and RRT tasks. The iModel's performance is measured by the seven effectiveness metrics and compared with its original performance before the integration with the UR method. The statistical significance tests, the percentage change and the t-test, are used to measure the improvement in the iModel performance and verify whether it is statistically different from the original's. If the new performance is significantly better than the original one, then we can claim that the UR method can reduce uncertainties and the evaluation hypothesis is valid.

All detailed results and comparisons are presented based on the experimental task and the type of the baseline model (i.e., supervised or unsupervised). Tables 6.10, 6.12, 6.13, 6.14 and 6.15 as well as Figures 6.16 and 6.14 illustrate all models results for the IF task while Table 6.11 and Figure 6.15 show the models performance in the RRT's experiments.

- **UR with Supervised Models**

  Eight supervised baseline models were used to evaluate the UR method for the IF and RRT tasks. These models are SVM, BM25, Prob, RFD$_2$, Rocchio, LASSO, Chi-square (χ^2) and MI. All these models adopt the low-level terms as text features except the RFD$_2$ that uses a combination of patterns and terms. For the IF results, the first eight rows of Table 6.10 shows each model performances before and after applying the UR method. The ”improvement%” row shows the percentage of improvement achieved by applying the UR method to the corresponding model’s original feature set. The table clearly shows that the re-ranking function of the UR method can significantly improve the performance of the feature set discovered by each model.

  As can be seen in Table 6.10, all eight models gained significant improvements in all the effectiveness measures. On an overall average across these measures, χ^2 achieved the highest improvement of 55.446% compared to its original performance in the IF task while the lowest improvement (only 5.931%) was obtained by the RFD$_2$ model. LASSO recorded the second highest improvement (50.521%) after the χ^2 followed by MI (48.771%), then BM25 (27.127%), Prob (27.060%) and lastly the SVM (24.262%) in descending order. Rocchio achieved a bit higher improvement (8.294%) than the RFD$_2$, which makes it the second lowest model to be improved by the UR method.

  While Table 6.10 showed the best results of these models using different top-k terms as queries to the IF system, Figure 6.14 shows the changes in MAP values for each model with an incremental change in the percentage of the top-k terms starting from top-1% to 100% of the entire terms space of each collection used by the model and averaged over the used 50 collections. It is apparent from the figure that the re-ranked term set performs significantly better at any percentage of terms in the original set, and usually, compared with the original term set, requires less re-ranked terms in numbers to obtain the highest performance. Moreover, the re-ranked terms showed significant performance stability and adequate sensitivity to the hyperparameter k compared to the original term sets. Similar figures for the P@20, BP, F$_{\beta=1}$ and IAP measures can be found in Appendix C.

  In the RRT task, all the eight models obtained significant improvements and outperformed their original performances, as illustrated in Table 6.11. The MI model achieved the highest average improvement (8027.121%) compared to its original nDCG value. Rocchio scored the lowest average improvement (20.876%) in the RRT task slightly proceeded by the RFD$_2$ at

21.976%. These results in Table 6.11 were obtained using the top-4 terms from each term set ranked by an original and improved model. However, Figure 6.15 shows the changes in the nDCG measure over the first 25 terms ($1 \leq k \leq 25$) in which all improved models performed significantly and consistently better than the originals.

The percentage of change and the t-test were also conducted in the UR method experiments in order to verify that the gained performances of the baseline models were statistically significant than their original ones. The percentage of change results in Tables 6.10 and 6.11 clearly show that all models improvements were higher than 5.0%, which implied that all improvements were statistically different from the original performances. The t-test results in Tables 6.12 and 6.11 confirmed the results of the percentage change. All p-values of all seven measures for both IF and RRT tasks were largely less than 0.05 in the two tails of the t-test, which indicate that all improvements were statistically significant. However, the RFD_2 model did not achieve an improvement that is higher than 5.0% on the $\text{F}_{\beta=1}$ measure ($3.784\% < 5.0\%$) in the IF results in Table 6.10 even though the t-test results of this measure in Table 6.12 indicated the opposite as the p-value at the two tails were less than 0.05 (0.003 and 0.005, respectively). Moreover, the two-tailed p-value of the t-test did not show that the improvement of the RFD_2 on the BP measure was statistically different from the original one. However, the one-tailed shows the opposite as the p-value (0.0372) is less than 0.05 and the percentage of change in Table 6.10 indicates that it is higher than 5.0% (5.496%).

- **UR with Unsupervised Models**

  Four unsupervised models were used in the experiments of the UR method to assess its effectiveness in reducing uncertainties. These TFS models are the pattern-based PDS, the topic-based LDA and its predecessor, the PLSA, and lastly the traditional TFIDF as a term-based method. These models, including the UR itself, were not trained on the negative document sets of the 50 RCV1 collections. The models' results for the IF task are presented in Table 6.10 at its last four rows. By examining these rows, we can see that the UR method can significantly improve the performances of these models in all measures.

  In Table 6.10, PLSA achieved the highest overall average improvement of 29.604% across all measures. The term-based TFIDF scored the second best improvement (15.990%) followed by the LDA (14.107%) and lastly the PDS with an overall average improvement of 12.841%. These results were the best results for each model, and they were achieved using different top-k terms based on the model's ability to respond to the scaling function of the UR method.

Therefore, for a much clearer picture, Figure 6.14 illustrates the response of each model to the UR method for the entire terms space of the model measured by the MAP metric. We can see that at each top-$k\%$ of the terms the improved model achieved much higher performance compared to the original model and at a much smaller number of terms. Even more, each improved model showed much performance stability and adequate sensitivity to the k hyperparameter. While this figure shows the MAP results, similar figures for the other effectiveness measures can be located in Appendix C.

In the RRT experiments, the last four rows of Table 6.11 show the results of the used unsupervised models measured by the nDCG metric. It is apparent that applying the UR method to these models made them perform very effectively compared to their original performances in the RRT task. TFIDF re-ranked terms scored the best average improvement of 361.903$\%$ compared to their original performance (0.117 $\gg$ 0.025). The re-ranked topical terms of the PLSA model also achieved significant improvement compared to its original performance by an average of 77.771$\%$ while the improved terms of its successor, the LDA, only gained 28.903$\%$ improvement. The re-ranked terms of the PDS patterns scored the best nDCG result (0.490) among all baseline models with an average improvement of 43.418$\%$ compared to the original model's performance. Although these results were scored using the top-4 re-ranked terms of each model, Figure 6.15 shows that all the improved models performed consistently much better than the originals at any top-k value of the first 25 words.

The percentage of change results in Tables 6.10 and 6.11 as well as the t-test results in Tables 6.12 and 6.11 strongly confirm that all the reported performances improvements of these unsupervised models are statistically different from their original performances. As shown in these tables, all improvement$\%$ of the percentage change test were much higher than 5.0$\%$ in all measures. Similarly, for the t-test, all p-values of all tails were much less than 0.05.

- **Best Model Versus All Models**

  The previous sections presented the improvement gains in both supervised and unsupervised baseline models after applying the UR method. This section presents the results of the best-improved model (i.e., iModel) compared to the other models for the same IF and RRT tasks. These results are shown in Tables 6.13, 6.14, 6.15 and Figure 6.16 for all tasks.

  The best performance in the IF task was scored by the improved SVM model (i.e., iSVM) as shown in Table 6.13. Compared to the second best-improved model, the iBM25, iSVM

Table 6.10: The performance improvements of all TFS models for the IF task after applying the UR method compared to their original performance averaged over the first 50 collections of the RCV1 dataset

Model	P@20	BP	MAP	$F_{\beta=1}$	IAP
SVM	0.491	0.414	0.436	0.437	0.462
iSVM	0.613	0.531	0.559	0.502	0.578
improvement%	+**24.847**%	+**28.442**%	+**28.178**%	+**14.817**%	+**25.025**%
BM25	0.445	0.407	0.407	0.414	0.428
iBM25	0.596	0.526	0.553	0.504	0.570
improvement%	+**33.933**%	+**29.238**%	+**35.872**%	+**21.739**%	+**33.178**%
Prob	0.464	0.395	0.414	0.421	0.438
iProb	0.593	0.515	0.542	0.499	0.559
improvement%	+**27.802**%	+**30.486**%	+**30.751**%	+**18.467**%	+**27.794**%
RFD_2	0.525	0.461	0.474	0.459	0.497
$iRFD_2$	0.563	0.487	0.506	0.476	0.529
improvement%	+**7.238**%	+**5.496**%	+**6.598**%	+**3.784**%	+**6.540**%
Rocchio	0.509	0.430	0.456	0.446	0.480
iRocchio	0.559	0.469	0.496	0.469	0.521
improvement%	+**9.823**%	+**8.927**%	+**8.847**%	+**5.333**%	+**8.541**%
LASSO	0.329	0.325	0.318	0.354	0.347
iLASSO	0.565	0.467	0.495	0.468	0.516
improvement%	+**71.733**%	+**43.663**%	+**55.995**%	+**32.296**%	+**48.920**%
χ^2	0.316	0.309	0.304	0.346	0.329
iχ^2	0.541	0.467	0.492	0.472	0.514
improvement%	+**71.203**%	+**51.139**%	+**62.153**%	+**36.389**%	+**56.348**%
MI	0.328	0.319	0.309	0.344	0.341
iMI	0.545	0.458	0.476	0.460	0.498
improvement%	+**66.159**%	+**43.705**%	+**54.335**%	+**33.544**%	+**46.113**%
PDS	0.496	0.430	0.444	0.439	0.464
iPDS	0.574	0.489	0.526	0.483	0.549
improvement%	+**15.726**%	+**13.721**%	+**18.468**%	+**10.023**%	+**18.319**%
LDA	0.492	0.414	0.442	0.437	0.468
iLDA	0.565	0.483	0.512	0.479	0.532
improvement%	+**14.930**%	+**16.605**%	+**15.800**%	+**9.576**%	+**13.623**%
PLSA	0.423	0.386	0.379	0.392	0.404
iPLSA	0.582	0.478	0.509	0.478	0.528
improvement%	+**37.589**%	+**23.834**%	+**34.301**%	+**21.939**%	+**30.693**%
TFIDF	0.354	0.338	0.337	0.367	0.366
iTFIDF	0.458	0.381	0.390	0.399	0.415
improvement%	+**29.379**%	+**12.723**%	+**15.768**%	+**8.765**%	+**13.317**%

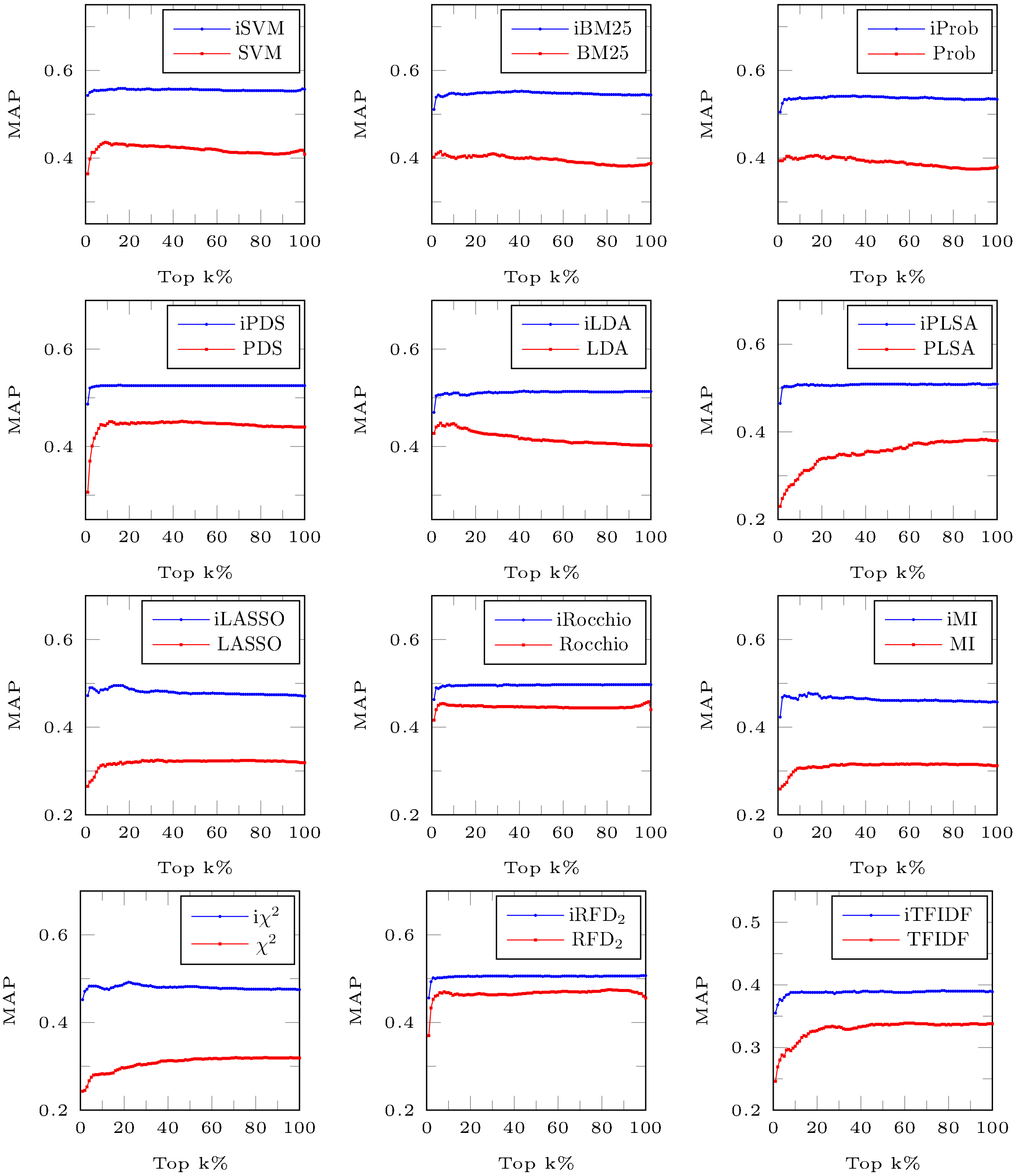


Figure 6.14: The changes in the MAP measure for each TFS model before and after applying the UR method for the IF task using top 1% to 100% of the terms space of each collection averaged over all 50 collections.

Table 6.11: The performance improvement of all TFS models for the RRT task after applying the UR method compared to their original performance averaged over the first 50 collections of the RCV1 dataset

Model	nDCG@4	improvement%	T-Test p-value	
SVM	0.058	+**621.469**%	One-Tailed	5.882E-13
iSVM	**0.422**		Two-Tailed	1.176E-12
BM25	0.083	+**321.618**%	One-Tailed	3.220E-09
iBM25	**0.351**		Two-Tailed	6.440E-09
Prob	0.060	+**470.657**%	One-Tailed	7.073E-10
iProb	**0.345**		Two-Tailed	1.415E-09
RFD$_2$	0.355	+**21.976**%	One-Tailed	2.129E-04
iRFD$_2$	**0.433**		Two-Tailed	4.258E-04
Rocchio	0.330	+**20.876**%	One-Tailed	2.983E-04
iRocchio	**0.399**		Two-Tailed	5.966E-04
LASSO	0.007	+**5680.761**%	One-Tailed	2.186E-14
iLASSO	**0.428**		Two-Tailed	4.372E-14
χ^2	0.009	+**3251.463**%	One-Tailed	9.265E-10
iχ^2	**0.315**		Two-Tailed	1.853E-09
MI	0.004	+**8027.121**%	One-Tailed	2.511E-13
iMI	**0.329**		Two-Tailed	5.021E-13
PDS	0.342	+**43.418**%	One-Tailed	2.298E-05
iPDS	**0.490**		Two-Tailed	4.596E-05
LDA	0.356	+**28.903**%	One-Tailed	9.279E-05
iLDA	**0.459**		Two-Tailed	1.856E-04
PLSA	0.235	+**77.771**%	One-Tailed	2.070E-05
iPLSA	**0.418**		Two-Tailed	4.140E-05
TFIDF	0.025	+**361.903**%	One-Tailed	1.014E-04
iTFIDF	**0.117**		Two-Tailed	2.029E-04

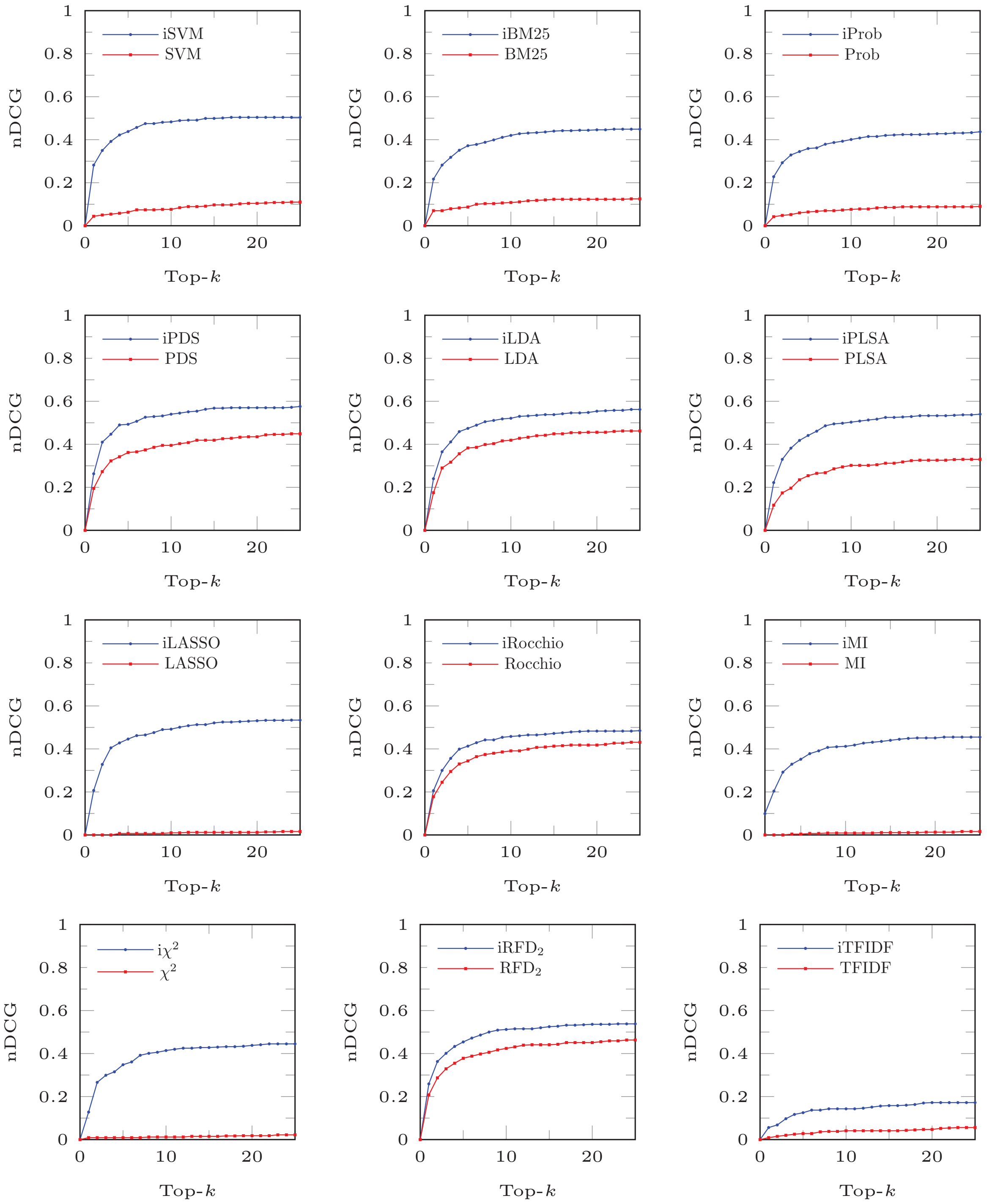


Figure 6.15: The changes in the nDCG@k measure for each TFS model before and after applying the UR method for the RRT task using the top 25 terms ($1 \leq k \leq 25$) averaged over the 50 human-assessed collections of the RCV1 dataset.

Table 6.12: The t-test p-values for each TFS model in comparison with its improved version after applying the UR method for the IF task results in Table 6.10

Model	Tail(s)	P@20	BP	MAP	$F_{\beta=1}$	IAP
SVM	One	2.275E-04	5.950E-06	1.509E-06	5.172E-06	7.557E-07
	Two	4.550E-04	1.190E-05	3.018E-06	1.034E-05	1.511E-06
BM25	One	1.098E-07	1.654E-09	1.258E-10	1.157E-10	4.135E-11
	Two	2.195E-07	3.307E-09	2.517E-10	2.313E-10	8.270E-11
Prob	One	7.169E-06	3.123E-09	1.799E-09	3.145E-09	5.905E-10
	Two	1.434E-05	6.246E-09	3.598E-09	6.290E-09	1.181E-09
RFD_2	One	1.087E-03	3.723E-02	3.128E-03	2.524E-03	1.457E-03
	Two	2.174E-03	7.446E-02	6.257E-03	5.049E-03	2.913E-03
Rocchio	One	1.576E-03	2.052E-03	1.270E-04	5.256E-04	6.461E-05
	Two	3.153E-03	4.104E-03	2.539E-04	1.051E-03	1.292E-04
LASSO	One	1.610E-09	4.341E-08	1.115E-10	9.176E-10	1.284E-10
	Two	3.220E-09	8.683E-08	2.230E-10	1.835E-09	2.568E-10
χ^2	One	1.116E-08	5.550E-10	4.502E-12	6.049E-11	3.737E-12
	Two	2.231E-08	1.110E-09	9.004E-12	1.210E-10	7.474E-12
MI	One	8.130E-09	1.417E-08	6.982E-10	1.754E-09	1.128E-09
	Two	1.626E-08	2.834E-08	1.396E-09	3.507E-09	2.256E-09
PDS	One	7.058E-03	1.506E-02	1.333E-04	4.095E-04	2.691E-05
	Two	1.412E-02	3.012E-02	2.666E-04	8.190E-04	5.382E-05
LDA	One	1.867E-03	7.017E-06	2.513E-06	1.842E-06	3.113E-06
	Two	3.734E-03	1.403E-05	5.027E-06	3.684E-06	6.225E-06
PLSA	One	3.329E-07	5.528E-06	2.352E-08	3.542E-08	2.754E-08
	Two	6.657E-07	1.106E-05	4.705E-08	7.085E-08	5.508E-08
TFIDF	One	1.143E-05	1.762E-03	2.549E-05	3.482E-05	4.150E-05
	Two	2.286E-05	3.524E-03	5.098E-05	6.964E-05	8.299E-05

Table 6.13: The results of the improved TFS models for the IF task compared to the result of the best improved model (i.e., iSVM)

Model	P@20	BP	MAP	$F_{\beta=1}$	IAP
iSVM	**0.613**	**0.531**	**0.559**	**0.502**	**0.578**
iBM25	0.596	0.526	0.553	0.504	0.570
iProb	0.593	0.515	0.542	0.499	0.559
iPDS	0.574	0.489	0.526	0.483	0.549
iLDA	0.565	0.483	0.512	0.479	0.532
iPLSA	0.582	0.478	0.509	0.478	0.528
$iRFD_2$	0.563	0.487	0.506	0.476	0.529
iRocchio	0.559	0.469	0.496	0.469	0.521
iLASSO	0.565	0.467	0.495	0.468	0.516
$i\chi^2$	0.541	0.467	0.492	0.472	0.514
iMI	0.545	0.458	0.476	0.460	0.498
iTFIDF	0.458	0.381	0.390	0.399	0.415
improvement%	+**2.852**%	+**0.976**%	+**1.215**%	−**0.440**%	+**1.419**%

Table 6.14: The results of iSVM model for the IF task compared to other TFS models as baselines (grouped as supervised and unsupervised)

Model	P@20	BP	MAP	$F_{\beta=1}$	IAP
iSVM	**0.613**	**0.531**	**0.559**	**0.502**	**0.578**
RFD_2	0.525	0.461	0.474	0.459	0.497
Rocchio	0.509	0.430	0.456	0.446	0.480
Prob	0.464	0.395	0.414	0.421	0.438
BM25	0.445	0.407	0.407	0.414	0.428
LASSO	0.329	0.325	0.318	0.354	0.347
MI	0.328	0.319	0.309	0.344	0.341
χ^2	0.316	0.309	0.304	0.346	0.329
improvement%	+**16.762**%	+**15.154**%	+**17.875**%	+**9.231**%	+**16.443**%
PDS	0.496	0.430	0.444	0.439	0.464
LDA	0.492	0.414	0.442	0.437	0.468
PLSA	0.423	0.386	0.379	0.392	0.404
TFIDF	0.354	0.338	0.337	0.367	0.366
improvement%	+**23.589**%	+**23.488**%	+**25.901**%	+**14.351**%	+**24.569**%

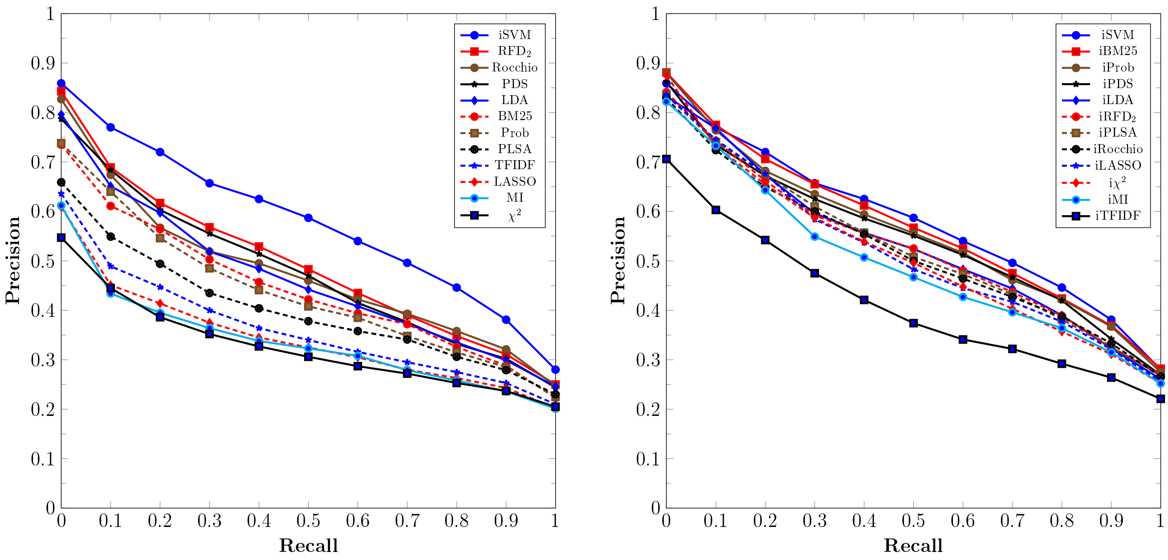


Figure 6.16: The 11-point result of the iSVM model for the IF task in comparison with other TFS models (left) and iSVM compared to other improved models (right) all averaged over the first 50 collections of the RCV1 dataset.

Table 6.15: The improved TFS models results for the RRT task compared to the result of the best improved model (i.e., iPDS)

Model	nDCG@4	improvement%
iPDS	**0.490**	0%
iLDA	0.459	+**6.754**%
iRFD$_2$	0.433	+**13.164**%
iLASSO	0.428	+**14.486**%
iSVM	0.422	+**16.114**%
iPLSA	0.418	+**17.225**%
iRocchio	0.399	+**22.807**%
iBM25	0.351	+**39.601**%
iProb	0.345	+**42.029**%
iMI	0.329	+**48.936**%
iχ^2	0.315	+**55.556**%
iTFIDF	0.117	+**318.803**%

achieved an overall average improvement of 1.204% nearly in all measures. iSVM scored its highest average improvement on the P@20 measure while its lowest was -0.440% on the $F_{\beta=1}$ metric. These improvements indicate that iSVM performance is not significantly different from the iBM25 in IF. Figure 6.16 (right) confirms this conclusion as these two models performed similarly well in IF. In the RRT task, the iPDS model achieved the best result of 0.490 on the nDCG measure, as illustrated in Table 6.15, with average improvements of 6.754% and 13.164% compared to the iLDA and iRFD$_2$, respectively.

Table 6.14 and Figure 6.16 (left) compare the performance of iSVM with all the baseline models. The improvement% at the bottom of Table 6.14 shows the percentage of improvement achieved by iSVM against the best-supervised baseline model, RFD$_2$, and the best-unsupervised baseline model, PDS. The iSVM model outperforms all models in all five measures. The improvement of iSVM against the RFD$_2$ model is from a maximum of 17.875% to a minimum of 9.231% in all measures. The iSVM model also outperformed the PDS by a maximum improvement of 25.901% and a minimum of 14.351%. The performance improvements against the most important measure for the IF system, MAP, are 17.875% and 25.901% compared to RFD$_2$ and PDS, respectively. Generally, iSVM achieved average improvements of 15.093% and 22.380% in all measures against RFD$_2$ and PDS, respectively. The interpolated precision results of 11 standard recall levels in Figure 6.16 (left) show that iSVM consistently outperforms any baseline models.

Based on all results presented in previous sections, we can conclude that when our UR method is applied to suitable relevant feature discovery model, the performance can be significantly better than existing models. Therefore, all these results support our hypothesis that paragraph relevance can effectively reduce uncertainties in relevant feature space.

Apart from the effectiveness of SIF, SIF2 and the UR method so far, they (1) cannot deal with the problem of unbalanced relevant topics in order to select the most relevant features. Also, as they are unsupervised, they (2) cannot select relevant features that frequently occur in negative documents. In the following sections, the results of the proposed USIF and SSIF frameworks will be presented. USIF was developed to address the first problem while SSIF was proposed to deal with the second problem. Both frameworks make use of SIF and the UR methods in different ways to deal with uncertainties inherited in relevance feedback.

6.8.4 The Proposed USIF Framework

USIF results for the IF testing system are illustrated in Table 6.16 and Figure 6.17 (left) and compared to the results of different baseline models for the same task. The baseline models are grouped based on the fusion strategy they use as either early or late or hybrid. Table 6.17 and Figure 6.17 (right) show USIF results for the RRT task including the baselines. The percentage change and the t-test results are presented in Tables 6.16, 6.18 and 6.17 showing the statistical significance of USIF performance in both experimental tasks against the best baseline models. More evaluation details are described below.

- **USIF Versus Early Fusion Models**

  USIF performance for IF was compared to Rocchio, SVM and BM25 as early fusion-based models. Unlike USIF, they are supervised and use low-level terms. Table 6.16 shows these models performances, including USIF, in the upper part of the table measured by the five standard metrics; P@20, BP, MAP, F$_{\beta=1}$ and IAP. The improvement% row at the bottom of this part shows the percentage of improvement achieved by the USIF against the best model (i.e., Rocchio) among all the other baseline models in that part. Figure 6.17 (left) illustrates the performance of USIF and these baseline models for the same IF task measured by the 11-point metric.

  It is apparent from the first part of Table 6.16 and Figure 6.17 that the USIF consistently performs the best among all these early-fusion models. It outperformed Rocchio's performance by an average improvement of 18.611% across all measures with a maximum improvement of 21.022% on P@20 and a minimum of 12.290% on $F_{\beta=1}$. Moreover, USIF was significantly better than Rocchio on the 11-point measure, as illustrated in Figure 6.17 (left). For the RRT task, our framework scored an average of 0.502 on nDCG@4 that indicates that USIF was significantly better than Rocchio by an average improvement of 52.112% as shown in Table 6.17.

USIF's score on the nDCG@4 measure is the best score achieved among all TFS models used in this research, including our proposed works. While the score was achieved using four terms discovered by USIF, Figure 6.17 (right) shows that our framework was consistently better than all baseline models, including Rocchio. All USIF improvements were statistically significant compared to the baselines as confirmed by the percentage change and t-test results in Tables 6.16, 6.17 and 6.18. USIF improvements were much higher than 5.0% and their t-test p-values were largely less than 0.05 in both tails of the test.

Table 6.16: The USIF results for the IF task compared to the baselines (grouped based on the fusion strategy they use to early, late and hybrid fusion models) for all measures averaged over the first 50 document collections of the RCV1 dataset

Model	P@20	BP	MAP	$F_{\beta=1}$	IAP
USIF	**0.616**	**0.518**	**0.550**	**0.500**	**0.571**
Rocchio	0.509	0.430	0.456	0.446	0.480
SVM	0.491	0.414	0.436	0.437	0.462
BM25	0.445	0.407	0.407	0.414	0.428
improvement%	+**21.022**%	+**20.285**%	+**20.638**%	+**12.290**%	+**18.822**%
LDA	0.492	0.414	0.442	0.437	0.468
SCSP	0.480	0.407	0.420	0.423	0.442
TNG	0.447	0.360	0.372	0.386	0.394
LdaConcept	0.335	0.329	0.326	0.352	0.357
improvement%	+**25.305**%	+**25.063**%	+**24.403**%	+**14.428**%	+**21.859**%
RFD_2	0.561	0.473	0.493	0.470	0.514
PDS	0.496	0.430	0.444	0.439	0.464
MP	0.426	0.392	0.393	0.409	0.421
improvement%	+**9.804**%	+**9.506**%	+**11.477**%	+**6.504**%	+**11.110**%

- **USIF Versus Late Fusion Models**

  Four baseline models were selected to represent this category. They are the supervised pattern-based SCSP model and three unsupervised models, which includes the topic-based LDA, the topical phrase-based TNG and the topical concept-based LdaConcept. The second part of Table 6.16 shows the IF results of these baseline models in descending order and

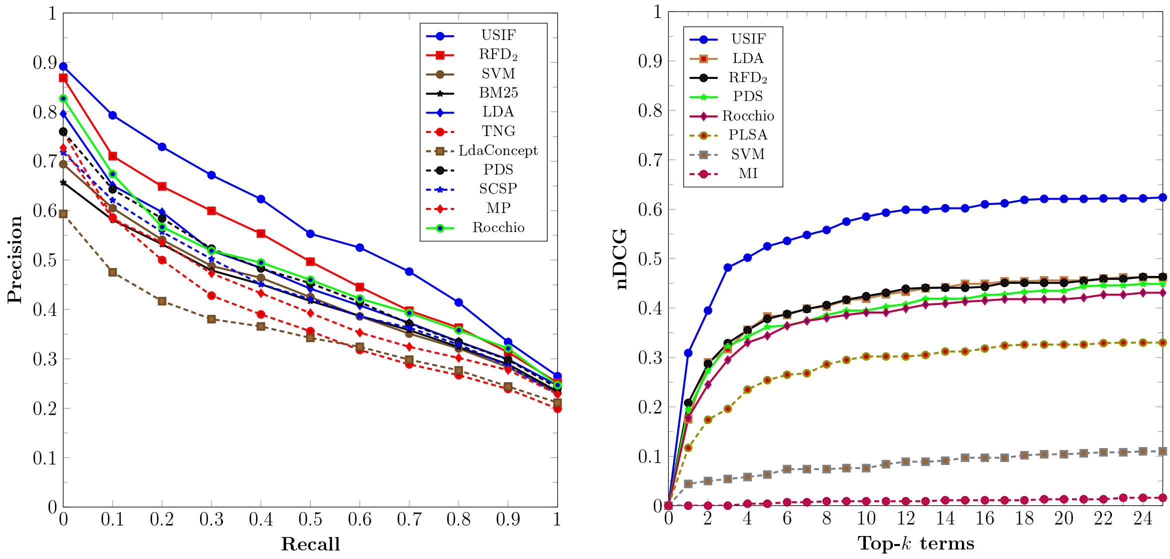


Figure 6.17: The 11-point results for IF (left) and the nDCG@k results for RRT (right) of USIF in comparison with baselines averaged over the first 50 collections of the RCV1 dataset.

Table 6.17: The USIF results for the RRT task including the percentage change and the t-test p-value in comparison with some of the baselines averaged over the first 50 collections of the RCV1 dataset

Model	nDCG@4	improvement%	p-value
USIF	**0.502**	0%	N/A
LDA	0.356	+**40.788**%	4.486E-06
RFD$_2$	0.355	+**41.371**%	3.009E-06
PDS	0.342	+**46.726**%	9.198E-06
Rocchio	0.330	+**52.112**%	4.717E-07
SVM	0.058	+**758.720**%	2.288E-14

Table 6.18: The t-test p-values of the best baseline model in each category in comparison with the USIF framework for the IF task tesults in Table 6.16

Model	Tail(s)	P@20	BP	MAP	F$_{\beta=1}$	IAP
Rocchio	One	1.594E-06	1.603E-05	9.837E-07	5.877E-06	7.206E-07
	Two	3.188E-06	3.205E-05	1.967E-06	1.175E-05	1.441E-06
LDA	One	1.901E-05	1.783E-08	2.999E-08	1.534E-07	7.729E-09
	Two	3.801E-05	3.567E-08	5.999E-08	3.068E-07	1.546E-08
RFD$_2$	One	2.458E-02	7.596E-03	1.528E-03	3.381E-03	6.739E-04
	Two	4.916E-02	1.519E-02	3.056E-03	6.761E-03	1.348E-03

USIF was compared to the best model; the LDA. The performance of USIF is significantly better than the performance of LDA by an overall average improvement of 22.212% across all measures. The performance improvement by USIF is from an average minimum of 14.428% on the $F_{\beta=1}$ measure to a maximum of 25.305% when compared with LDA. The 11-point results in Figure 6.17 show that the performance of USIF is consistently better than the LDA.

In the RRT experiment, Table 6.17 shows that USIF performance was superior to the LDA by an average improvement of 40.788% on the nDCG@4 measure, and Figure 6.17 (right) confirms the superiority of USIF at any number of terms ranging from the top-1 to top-25 keywords. All USIF improvements compared to the baselines were not random as verified by the statistical significance tests. The percentage change measure and the t-test results in Tables 6.16, 6.17 and 6.18 show that all improvements in USIF performance were statistically different from the baselines as they were much higher than 5.0%. The two tails t-test confirm the results of the percentage change as all p-values were significantly less than 0.05.

- **USIF Versus Hybrid Fusion Models**

  The last part of Table 6.16 shows the IF experimental results of the USIF framework compared to three hybrid fusion-based baseline models. These models fuse high-level patterns with low-level terms, and they are the supervised RFD_2 model, and the unsupervised PDS and MP models. USIF performances were compared against RFD_2, as the best model in the group. USIF outperformed RFD_2 by an overall average improvement of 9.680% in all measures. The maximum average improvement achieved by USIF was 11.477% on the MAP metric, and the minimum was 6.504% on the $F_{\beta=1}$ measure. Moreover, Figure 6.17 (left) shows that USIF continues to perform significantly better than RFD_2 on the 11-point metric.

  USIF was superior to RFD_2 in discovering relevant terms, as illustrated in Table 6.17. Our framework achieved an average improvement of 41.371% compared to the RFD_2 and consistently superior not only using four terms but at any number of the first 25 words as can be seen in Figure 6.17 (right). USIF achievements against RFD_2 in both IF and RRT tasks were also statistically verified using the percentage of change and t-test to make sure that they did not occur randomly. All USIF improvements were higher than 5.0% in all seven measures as illustrated in Tables 6.16 and 6.17. The two tails t-test results in Tables 6.18 and 6.17 strongly confirm the results of the percentage of change as all p-values were much less than 0.05.

As per the results presented earlier, we have much confidence in claiming that the USIF framework can discover relevant features from a set of unbalanced latent topics that discuss user information preferences. USIF managed to effectively select and weight these features using a combination of unsupervised learning algorithms and representative global statistics. Therefore, the experimental results discussed above support hypothesis 4.

While USIF provided a comprehensive, unsupervised solution for discovering relevant features from a set of positive documents, it still cannot deal with relevant features that also frequently appear in negative documents. The following section presents the results of our SSIF framework that effectively addresses the limitation of USIF.

6.8.5 The Proposed SSIF Framework

The performances of SSIF in IF and RRT experiments are presented in this section and compared to different state-of-the-art baseline models. We grouped supervised models together, including SSIF for easier comparison. Also, we group other baseline models based on the learning or mining algorithms they use for better analysis. The results of SSIF and the baseline models for IF are given in Table 6.19 and Figure 6.18 (left) while their results in RRT are illustrated in Table 6.20 and Figure 6.18 (right). We also conducted two statistical tests; the percentage change and t-test, to measure and verify how significant the SSIF improvements compared to the baselines. These tests results are presented in Tables 6.19, 6.20 and 6.21. It is apparent from all these tables and figures that the SSIF consistently performs the best among all baseline models. More detailed comparisons are given below in the following sections.

- **Comparison with Supervised Learning**

  For IF, the first part of Table 6.19 shows that the SSIF outperformed all other supervised learning-based baseline models in all five measures. The improvement% in this part shows that SSIF, which combines both supervised and unsupervised learning, consistently achieved the best performance when compared with baseline models that are based on supervised learning. The improvement of SSIF against the second best model, RFD$_2$, was from a minimum of 9.519% to a maximum of 16.880% on F$_{\beta=1}$ and MAP measures, respectively. The performance improvement against the most important measure of IF system, MAP, was 16.880%, and the average improvement in all five measures was 14.108%. The 11-point results in Figure 6.18 (left) clearly shows that SSIF performance was significantly better than RFD$_2$ and all other models.

  Table 6.20 shows SSIF and other suitable models performances in the RRT task using the first top four terms. SSIF scored 0.420 on the nDCG measure with an average improvement of 18.292% compared to RFD_2. For the same task, SSIF continued to perform consistently better than RFD_2 at different k values, as shown in Figure 6.18 (right). Moreover, the percentage change results in Tables 6.19 and 6.20 show that all SSIF improvements against RFD_2 were statistically significant as they were largely higher than 5.0% in all measures. T-test results in Tables 6.21 and 6.20 further confirm the statistical significance of SSIF performance compared to RFD_2. All p-values were much less than 0.05 in all measures except for the nDCG@k. It was not statistically different from RFD_2 result ($0.113 \not< 0.05$).

Table 6.19: The SSIF results for the IF task compared to the baselines for all measures averaged over the first 50 document collections of the RCV1 dataset

Model	P@20	BP	MAP	$F_{\beta=1}$	IAP
SSIF	**0.631**	**0.550**	**0.576**	**0.515**	**0.592**
RFD_2	0.561	0.473	0.493	0.470	0.514
SVM	0.491	0.414	0.436	0.437	0.462
BM25	0.445	0.407	0.407	0.414	0.428
improvement%	+**12.478**%	+**16.405**%	+**16.880**%	+**9.519**%	+**15.256**%
LDA	0.492	0.414	0.442	0.437	0.468
PLSA	0.423	0.386	0.379	0.392	0.404
TNG	0.447	0.360	0.372	0.386	0.394
improvement%	+**28.356**%	+**32.942**%	+**30.433**%	+**17.667**%	+**26.406**%
PDS	0.496	0.430	0.444	0.439	0.464
SCSP	0.480	0.407	0.420	0.423	0.442
PCM	0.437	0.372	0.381	0.397	0.406
n-grams	0.401	0.342	0.361	0.386	0.384
improvement%	+**27.218**%	+**27.863**%	+**29.893**%	+**17.174**%	+**27.549**%
MPBTM	0.552	0.466	0.477	0.459	0.496
SPBTM	0.527	0.448	0.456	0.445	0.478
PBTM-FP	0.470	0.402	0.427	0.423	0.449
PBTM-FCP	0.489	0.420	0.423	0.422	0.447
improvement%	+**14.312**%	+**18.058**%	+**20.762**%	+**12.218**%	+**19.341**%

- **Comparison with Topic Modelling**

  The performance of the topic modelling-based baseline models in IF are shown in the second part of Table 6.19. The best model in this part is LDA. The performance of SSIF was significantly better than the performance of topic modelling-based baselines. The performance improvement by SSIF was from a minimum average of 17.667% to a maximum of 32.9% when compared with LDA based on the $F_{\beta=1}$ and BP measures, respectively. SSIF maintained an overall average improvement of 27.161% across all measures. The 11-point result in Figure 6.18 (left) confirms SSIF superiority over LDA and other topic-based models. In RRT, SSIF

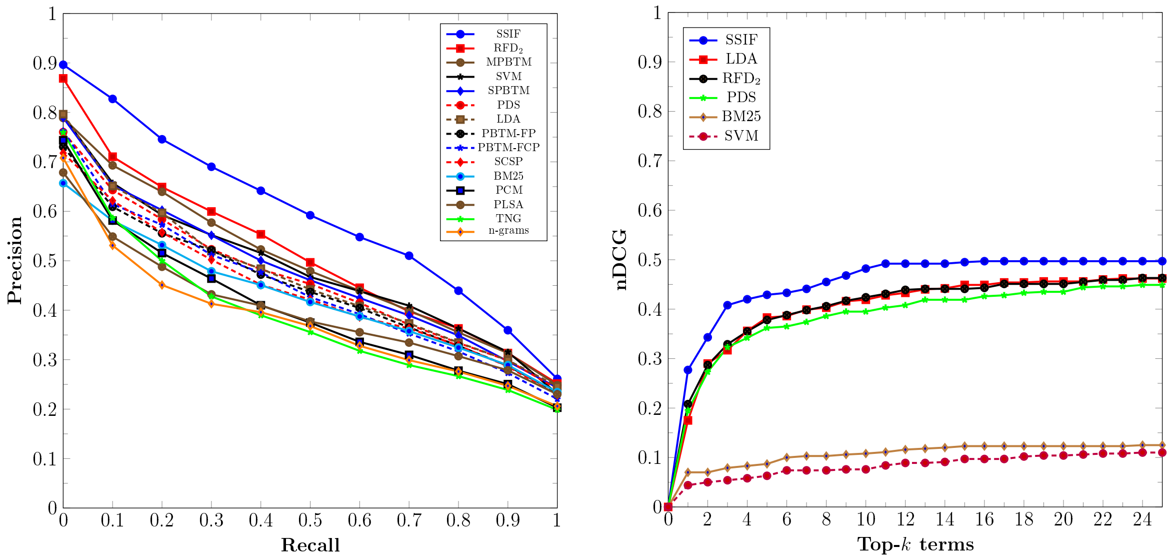


Figure 6.18: The 11-point results for IF (left) and the nDCG@k results for RRT (right) of SSIF in comparison with baselines averaged over the first 50 collections of the RCV1 dataset.

Table 6.20: The SSIF results for the RRT task including the percentage change and the t-test p-value in comparison with some of the baselines averaged over the first 50 collections of the RCV1 dataset

Model	nDCG@4	improvement%	p-value
SSIF	**0.420**	0%	N/A
LDA	0.356	+**17.978**%	1.367E-01
RFD_2	0.355	+**18.292**%	1.133E-01
PDS	0.342	+**22.807**%	8.610E-02
BM25	0.083	+**403.974**%	4.187E-10
SVM	0.058	+**618.533**%	3.321E-11

Table 6.21: The t-test p-values of the best baseline model in each category in comparison with the SSIF framework for the IF task results in Table 6.19

Model	Tail(s)	P@20	BP	MAP	$F_{\beta=1}$	IAP
RFD_2	One	5.463E-03	3.574E-04	2.866E-04	6.626E-04	2.347E-04
	Two	1.093E-02	7.148E-04	5.733E-04	1.325E-03	4.693E-04
LDA	One	1.095E-05	1.132E-07	1.857E-07	5.673E-07	2.076E-07
	Two	2.191E-05	2.264E-07	3.715E-07	1.135E-06	4.153E-07
PDS	One	1.787E-06	4.254E-06	3.663E-07	8.594E-07	2.045E-07
	Two	3.574E-06	8.509E-06	7.325E-07	1.719E-06	4.089E-07
MPBTM	One	3.208E-03	7.582E-04	7.677E-05	2.075E-04	5.350E-05
	Two	6.417E-03	1.516E-03	1.535E-04	4.149E-04	1.070E-04

achieved better performance than LDA with an average improvement of 17.978% over LDA performance on the nDCG@4 measure. Figure 6.18 (right) also illustrates SSIF performance in RRT over LDA using the top 25 terms ranked by both SSIF and LDA.

SSIF improvements in all measures for the experimental task were statistically significant from the LDA as measured by the percentage change test. The percentage change results in Tables 6.19 and 6.20 clearly indicate that SSIF improvements were much higher than 5.0%. The t-test confirmed the results of the percentage change. The p-values of both tails of the test show that SSIF results were statistically different from the LDA as their p-values were much less than 0.05, as illustrated in Tables 6.21 and 6.20. However, this was not the case with the nDCG@4 result because the t-test p-value of SSIF compared to LDA was higher than 0.05 indicating that SSIF performance in RRT was not statistically significant than the LDA's negating the outcome of the percentage change test.

- **Comparison with Pattern Mining and N-Grams**

  The third part of Table 6.19 shows the performance of pattern mining-based baseline models in IF including the phrase-based N-Grams model. The best model in this part is PDS. The minimum and maximum improvements achieved by the SSIF against PDS is 17.174% and 29.893% on the $F_{\beta=1}$ and MAP measures, respectively. Over all measures, SSIF performance was significantly better than PDS by an average improvement of 25.939%. The 11-point results in Figure 6.18 (left) confirm the previous overall average improvement of SSIF over the PDS model in the IF task. SSIF also continued to outperform PDS in the RRT experiment achieving an average improvement of 22.807% over it on the nDCG@k measure, as shown in Table 6.20.

  Figure 6.18 (right) shows that SSIF scored better results on nDCG not only when k=4 but at any k value from 1 to 25. All SSIF improvements over PDS were statistically significant as indicated by the percentage of change measure. Tables 6.19 and 6.20 clearly show that SSIF performance improvements in both experimental tasks were largely higher than 5.0%. The two-tailed t-test confirmed the outcome of the percentage change. Nearly all p-values of the two tails of the test were much less than 0.05 as can be seen in Tables 6.21 and 6.20 except for the nDCG@4 result. Its p-value was slightly higher than 0.05 refuting the outcome of the percentage change (22.807%), which stated that SSIF improvement was statistically significant than PDS.

- **Comparison with Topical Pattern Mining**

  The last part of Table 6.19 shows the performance of topical pattern mining-based baseline models. As these models combine the best of both topic modelling and pattern mining, they outperform the models in the second part and the third part. The performances of the models in this part are about the same as the models in the first part except for SSIF. The best-performing model in this part is MPBTM. The improvement of SSIF against MPBTM in IF is from a minimum of 12.218% to a maximum of 20.762% on the $F_{\beta=1}$ and MAP measures, respectively.

  Across all five measures, SSIF outperformed MPBTM by an average improvement of 16.938%. The 11-point results in Figure 6.18 (left) can confirm this improvement in which SSIF maintained its superior performance over MPBTM. All SSIF improvements against MPBTM were statistically significant according to the percentage of change test. Its improvements were higher than 5.0% as demonstrated by the improvement% in Table 6.19. T-test results in Table 6.21 further confirm the statistical significance of SSIF improvements over the MPBTM model. All the p-values of the test were much less than 0.05 in all measures.

Based on the experimental results of the SSIF framework presented above, we are confident of claiming that SSIF can effectively select and weight relevant features that appear across both positive and negative documents. Our framework managed to do that through the combination of different supervised and unsupervised learning techniques. Consequently, those results presented earlier support hypothesis 5.

6.9 Analysis and Discussion

The previous section presented the extensive experimental studies that have been conducted to assess the effectiveness of our proposed TFS models and frameworks. The experimental results confirm the superiority of our techniques over all baseline models in both IF and RRT tasks. In this section, we further analyse and discuss these results based on the effects of some critical factors that influence the performance of our proposed models and frameworks as well as the used baselines. These factors are linked to the use of (1) fusion strategies; (2) type of text feature; (3) positive and/or negative feedback; and (4) global statistics. Also, the effects of other factors such as the sophistication of the weighting function and the learning algorithm are worth to be taken into consideration, especially when integrating different low-level and/or high-level features. A parameter-sensitivity analysis for SIF, SIF2 and UR models as well as a more in-depth investigation for the idea of separating feature selection from feature weighting in the proposed USIF and SSIF frameworks are also presented in this section.

6.9.1 The Proposed SIF Model

- **The Effects of Feature Type**

  As observed from the results shown in Tables 6.4 and 6.5 and illustrated in Figure 6.12, the SIF model outperformed all baseline models in all measures for both IF and RRT experimental tasks. Our SIF model achieved this superior performance through the hybrid fusion of high-level topics and low-level terms. Adopting only individual terms, as in BM25, TFIDF and other early fusion baseline techniques, made them performed poorly in IF compared to the late and hybrid fusion models. We speculate that the absence of semantic information among these terms is one of the main reasons behind the poor performance of these techniques despite the flexibility of terms and their rich statistical information. The inferior results of both TFIDF and BM25 in the RRT task, as illustrated in Table 6.5 and the right figure of Figure 6.12, evidently confirm the negative effects of ignoring semantic information in relevant feature discovery.

  However, (1) the efficient employment of the statistical properties of terms by the BM25 weighting function and, more specifically, (2) the utilisation of negative feedback made BM25 significantly better than TFIDF. Also, these two factors made BM25 competitive and sometimes even better, in some measures, than some of the late (e.g., PLSA, n-grams and TNG) and hybrid (e.g., MP) fusion-based models in the IF task, as illustrated in Table 6.4. Despite the positive effects of these factors on BM25, it still could not discover the relevant terms identified by the NIST experts, which made it performed very poorly in the RRT experiment. Table 6.22 shows a real example from Collection 101 of the RCV1 dataset in which BM25 could not highly rank any of these relevant terms (i.e., “Economic” and “Espionage”). Therefore, it is clear that the absence of semantic information in early fusion models has severely impacted their performance in IF and most apparently in the RRT task.

  The effective integration of topical features and the accurate estimation of their importance to some representative entities in relevant documents using multiple ERS have made our SIF model significantly better than LDA and PLSA. Both PLSA and LDA share a similar term weighting function, but LDA is more effective than PLSA in both IF and RRT experiments, as shown in Tables 6.4 and 6.5 as well as in Figure 6.12. It might be due to its underline Bayesian generative algorithm that can estimate more semantically related topical terms. All these models, including our SIF, utilise the semantic information that latent topics provide, and exploit the multi-topic assumption when representing relevant documents that discuss user information needs. However, the ERS-based weighting function of our SIF model assigns more accurate weights to topical terms than the LDA's. This claim can be testified by the performance of SIF in RRT, as illustrated in Table 6.5, and also can be seen clearly in Table 6.22 in which only SIF could automatically discover the human-identified relevant terms.

The adverse effects of (1) ignoring the multiple topics assumption in representing relevant documents; (2) the too strict constraint of the sequential appearance of terms in these documents; and (3) the ineffective term weighting functions have hindered the performance of phrase-, pattern-, and the hybrid feature-based models despite the semantic information in their high-level features. The negative effects of these three factors can be seen on the performance of the n-grams model, as illustrated in Table 6.4 and Figure 6.12. While the pattern-based MP and PDS models managed to solve the second factor, not dealing with the effects of the first and the third factors are clearly limiting their performance. However, the PDS model demonstrated significant performance compared to many baselines because it integrates the semantics of patterns with the statistical properties of low-level terms. It allows PDS to rank some relevant terms, as shown in Table 6.22, and be competitive with LDA.

The topical n-grams (TNG) model resolves the first factor, but clearly, not considering the effects of the second and the third factors badly influenced its performance. Despite dealing with the effects of the first two factors, the performance of the topical pattern-based models (i.e., PBTM-FP and PBTM-FCP) obviously impacted by the imprecision of their weighting functions. The proposed SIF model significantly outperformed all these models in all experimental tasks simply because SIF (1) represented the paragraphs of relevant documents with multiple topics; (2) relaxed the constraint of the sequential appearance of topical terms and (3) accurate estimated of the weight of these terms in the relevant documents that discuss what the user needs.

- **The Effects of Global Statistics**

  The proposed SIF model exploits the statistical properties of low-level terms, represented by the document frequency df, to estimate the relevance of topical terms at the collection

Table 6.22: The top-10 stemmed terms from collection 101 of the RCV1 dataset, which is about ’*economic espionage*’, discovered and ranked by different TFS models in which only SIF was able to select both these relevant features

SIF		**LDA**		**BM25**		**PDS**	
Term	*Weight*	*Term*	*Weight*	*Term*	*Weight*	*Term*	*Weight*
vw	0.423	piech	0.245	secret	0.130	vw	0.617
espionag	**0.236**	carmak	0.194	technolog	0.112	bill	0.343
piech	0.225	bill	0.185	crime	0.112	piech	0.340
year	0.221	feder	0.185	pass	0.112	men	0.256
bill	0.221	compani	0.180	fbi	0.098	**econom**	**0.238**
compani	0.216	men	0.171	bill	0.098	car	0.152
secret	0.194	photograph	0.145	cia	0.098	photograph	0.150
econom	**0.176**	camera	0.143	law	0.098	carmak	0.150
carmak	0.163	volkswagen	0.141	softwar	0.098	camera	0.134
feder	0.163	year	0.139	comput	0.098	volkswagen	0.125

level. However, several statistics can reveal the global importance of terms in a collection of relevant documents. Therefore, further experiments were conducted to measure the global informativeness of paragraph frequency pf and term frequency tf, as raw statistics of the individual terms in the collection. The popular hand-crafted statistics, namely the inverse-document frequency idf and the term frequency-inverse document $tfidf$, were also used in these experiments to measure their usability compared to pf and tf. Moreover, the experiments show how the inflexibility (e.g., low-frequency problem) of high-level features spaces (e.g., phrase space, pattern space, topic space, etc.) can be efficiently and effectively solved through the utilisation of the various statistics of the term space.

The experiments were conducted on the same 50 collections of the RCV1 dataset and for the same IF and RRT tasks. Table 6.23 and Figure 6.19 show the best results of the effects of used global statistics when integrated with SIF's weighting function (Equation 3.3). For IF, df remains the most representative global statistic when combined with SIF's equation. This combination scored an overall average improvement of 4.724$\%$ in all measures with a minimum improvement of 2.976$\%$ and a maximum of 5.587$\%$ in $F_{\beta=1}$ and P@20 measures, respectively, compared to the combination of the same equation with the second-best statistic, the paragraph frequency pf. The 11-point result exhibited on the left figure of Figure 6.19 shows that the combination with df obtained better precision scores than the combination pf at most of the 11-recall levels. All these improvements were achieved using only the top-$k = 10$ terms, as shown in Table 6.23, where the combination with pf required the next 33 terms (i.e., requires three times more terms than df) to score its best results in IF. In the RRT task, the same combination of df scored 0.457 on the nDCG@4 measure with an average improvement of 16.503% over pf's score (0.392) on the same measure. While the improvement obtained using top-4 terms, the right figure of Figure 6.19 illustrates that the combination of SIF's generalised weighting function and df is consistently better than the combination of the same equation with pf.

As seen from Table 6.23 and Figure 6.19, both pf and tf performed comparably similar when linearly integrated with SIF's weighing function. However, they were less effective in revealing the global importance of relevant topical terms compared to df as they might appear unevenly across the documents in the collection. The hand-crafted statistics, $tfidf$ and idf, performed very poorly on all measures for all tasks. Their performance was expected because they no longer resembled the original terms frequency, and were developed based on some assumptions to suit specific needs. Therefore, raw statistics of terms are more representative and can be used to resolve some frequency-based problems in high-level features.

Table 6.23: The IF and RRT results of SIF's main weighting function (Equation 3.3) integrated with different global statistics of low-level terms averaged over the 50 collections of the RCV1 dataset

	P@20	BP	MAP	$F_{\beta=1}$	IAP	nDCG@4	k
$pr(t) \times df(t)$	**0.567**	**0.475**	**0.500**	**0.473**	**0.527**	**0.457**	**10**
$pr(t) \times pf(t)$	0.537	0.452	0.478	0.459	0.500	0.392	43
$pr(t) \times tf(t)$	0.520	0.447	0.475	0.458	0.499	0.372	44
$pr(t) \times tfidf(t)$	0.406	0.357	0.361	0.380	0.390	0.069	49
$pr(t) \times idf(t)$	0.352	0.335	0.328	0.361	0.357	0.027	48
improvement%	+**5.587**%	+**5.075**%	+**4.684**%	+**2.976**%	+**5.297**%	+**16.503**%	

- **Parameters Sensitivity**

SIF uses two experimental parameters. The first is the number of LDA topics V and, as a hyperparameter, it can be difficult to be optimally set before training. The second parameter is the number of top relevant weighted terms k, which are used as a query to both IF and RRT testing system. Similar to V, it is challenging to know the optimal value for k from the data. Therefore, and to investigate the sensitivity of SIF to these two parameters, we conducted extensive experiments on the same RCV1 collections using all performance measures for the same experimental tasks.

The results of these experiments are presented in Figures 6.20 and 6.21. For IF, and using different values for V and k, our SIF model showed a very stable performance in all six

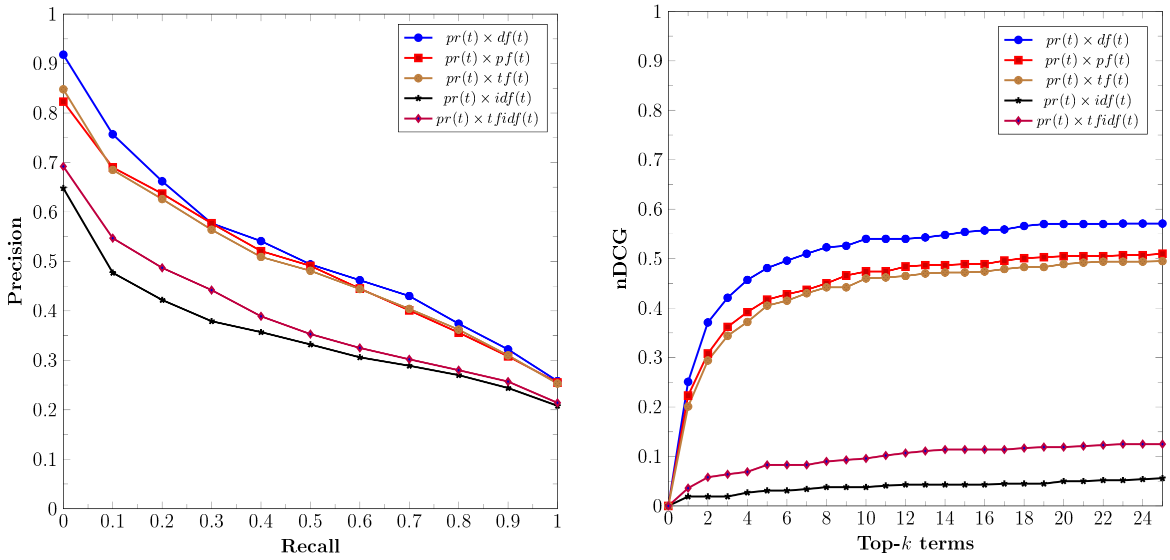


Figure 6.19: The results of 11-point measure (left) and nDCG at top-25 terms (right) of SIF's generalised weighting function (Equation 3.3) with other global statistics of terms averaged over the first 50 collections of the RCV1 dataset.

measures at any number of topics, as illustrated in the left figures of Figures 6.20 and 6.21. The model also demonstrated a stable performance after the first top ten terms ($k = 10$) in all measures except some slight fluctuations on the P@20 metric, as shown in the right figure of Figure 6.20. SIF also maintained the same stable performance in the RRT task. This can be seen in the right figure of Figure 6.21 where SIF obtained almost identical performance at any given value of the V and k parameters. Overall, despite the challenge of specifying optimal values for the V and k parameters, our SIF model is insensitive to these parameters, which gives it another significant advantage over many state-of-the-art TFS models of relevance discovery that might be sensitive to their experimental parameters.

6.9.2 The Proposed SIF2 Model

- **The Effects of Fusion Strategy**

The SIF model results that were presented and discussed in Sections 6.8.1 and 6.9.1 demonstrated the merits of adopting the hybrid fusion of high-level topics and low-level terms. The SIF2 model is regarded as an improved version of SIF. It continues to adhere to the same fusion strategy of SIF. However, SIF2 relaxes the constraint of SIF's assumption, which states that only one generalised score should be estimated and assigned to identical topical terms in each relevant document in the collection. This assumption means that each topical term in equally important to every document, which, in reality, might not be the case. To relax such assumption, SIF2 assumes that each topical term has specific local significance

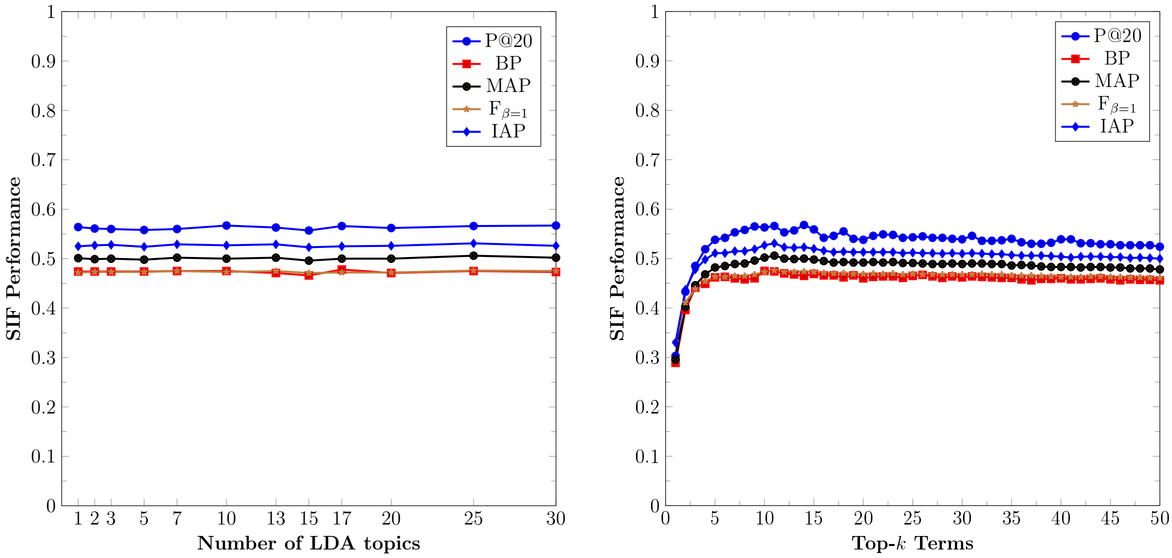


Figure 6.20: The SIF sensitivity to the number of LDA topics (left) and top-k terms (right) for the IF experiments.

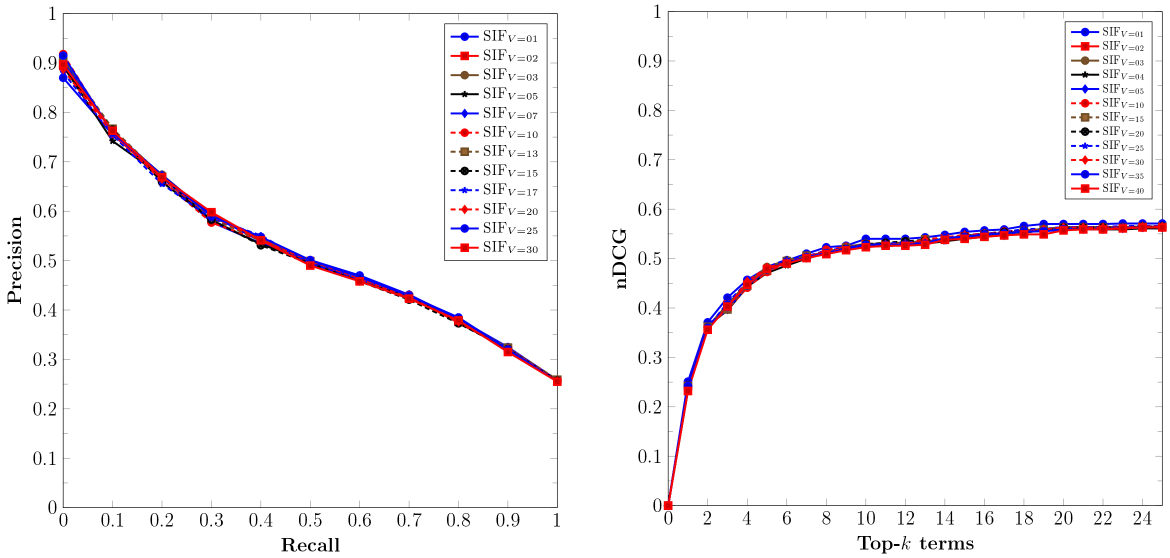


Figure 6.21: The SIF 11-point results for IF (left) and the SIF nDCG@k results for RRT (right) over different number of LDA topics.

at each document and has another global one at the collection level. The two significances must be integrated to represent the relevance of the term to the user information needs. The experimental results of SIF2 presented in Section 6.8.2 clearly demonstrated the superiority of SIF2 over the baseline models. Also, the comparison between SIF2 and SIF results described in Section 6.9.6 verified the validity of SIF2's assumption.

By adapting some of SIF's fusion steps and estimating more accurate weights for topical terms, SIF2 significantly outperformed both the supervised SVM and the unsupervised SPBTM models, as the best baseline TFS models in their categories. Despite the soundness of its mathematical foundation and the utilisation of negative documents, which made the SVM model performs better than many baselines, the model continues to show insufficient performance in selecting features for relevance discovery in accordance with the different studies in [Gao et al., 2015, Li et al., 2015, 2010, 2012, Zhong et al., 2012]. As an adherent of the early fusion strategy, the absence of semantics among the low-level terms used to represent documents for SVM apparently affected its performance. The late fusion-based SPBTM model was the best among the baselines due to the exploitation of the semantic information in the integrated representation of topics and patterns. However, the challenge of selecting the most important patterns extracted from relevant documents and ignoring the terms-topics distributions in these documents clearly hindered the SPBTM's performance compared to our SIF2 model. We continue to argue that assuming that only a particular group of patterns are important and ignoring others will lead to the loss of some relevant features.

Representing the paragraphs of relevant documents by multiple topics has made both SIF and SIF2 models performing effectively compared to those models that do not consider the topics in the paragraphs. Measuring the relevance of topical terms at the paragraph-level even improved the performance of LDA (LdaPara) in IF and RRT tasks compared to its performance at the document-level (LdaDoc), as can be seen in Tables 6.7 and 6.8 as well as in Figure 6.13. However, both, LdaPara and LdaDoc still could not estimate more accurate weights that reveal the relevance of topical terms for the reasons discussed previously. In the case of our SIF model, SIF2 revised SIF's weighting function and developed a more effective one that can go deeper into the structure of each relevant document and assign more accurate weights to topical terms. For example, in Table 6.24, it can be seen that only the SIF, SIF2 and PDS models can discover and highly rank human-identified relevant terms from Collection 101 of the RCV1 dataset.

Moreover, only our SIF and SIF2 models could discover the term ’*espionage*’ in which we argue that it is more topically specific and representative of the main topic of interest in Collection 101 than the word ’*economic*’. However, while both SIF and SIF2 ranked the word ’*espionage*’ as the second most relevant topical term, SIF2 estimated its relevance two times as much as SIF’s ($0.472 \gg 0.236$). As we mentioned in Section 6.4, we believe that these two words are not the only relevant terms in the collection, but to make the study simple and reliable, we only used those words identified by the NIST domain experts. Nevertheless, we argue that the top-10 terms of SIF2, shown in Table 6.24, are more meaningful and specifically relevant than SIFs. For example, the word ’*secret*’ is more relevant to the collection topic than ’*vw*’ (acronym of Volkswagen), which is highly ranked by SIF. Also, we can see that SIF2 was able to underestimate some general and frequent terms like ’*year*’ and ’*bill*’, and discover more specific ones to the context of ”Economic Espionage”, such as ’*trade*’ and ’*crime*’.

Table 6.24: The top-10 stemmed terms from collection 101 of the RCV1 dataset, which is about ’*economic espionage*’, discovered and ranked by different TFS models in which only SIF was able to select both these relevant features

SIF2		**SIF**		**LdaPara**		**SVM**		**PDS**	
Term	*Weight*	*Term*	*Weight*	*Term*	*Weight*	*Term*	*Weight*	*Term*	*Weight*
secret	0.709	vw	0.423	piech	0.245	vw	0.419	vw	0.617
espionag	**0.472**	**espionag**	**0.236**	carmak	0.194	piech	0.239	bill	0.343
compani	0.278	piech	0.225	bill	0.185	men	0.218	piech	0.340
trade	0.185	year	0.221	feder	0.185	bill	0.199	men	0.256
crime	0.179	bill	0.221	compani	0.180	photograph	0.175	**econom**	**0.238**
feder	0.164	compani	0.216	men	0.171	carmak	0.174	car	0.152
piech	0.139	secret	0.194	photograph	0.145	return	0.153	photograph	0.150
repres	0.121	**econom**	**0.176**	camera	0.143	volkswagen	0.130	carmak	0.150
volkswagen	0.108	carmak	0.163	volkswagen	0.141	gm	0.127	camera	0.134
pass	0.092	feder	0.163	year	0.139	camera	0.125	volkswagen	0.125

- **The Effects of Combining Local and Global Statistics**

  Unlike SIF, the SIF2 model considered the local statistics of a term t in each document using its paragraph frequency distribution. The paragraph distribution used to revise the term-topic distribution, which is globally estimated from the entire collection. By taking local details of topical terms into consideration, our SIF model effectively managed the hybrid fusion of high-level topics with both local and global statistics of low-level terms. Like the SIF model, SIF2 continues to use document frequency df to reveal the global relevance of the revised topical terms. Therefore, and, as other possible global statistics can be used for the same purpose of df, we conducted the same experiments of the effects of global statistics on the SIF model, which are discussed in Section 6.9.1, on Equation 4.4 of SIF2. For simplicity, we refer to the equation as $sr_{D^+}(t)$ instead of $\sum_{t \in d_i, d_i \in D^+} sr_{d_i}(t)$ in the table and figures of the results.

The experimental results presented in Table 6.25 and Figure 6.22 clearly show that df remains the most informative statistic for the global relevance of topical terms. In the IF task, and compared to the second-best results, the integration between Equation 4.4 and the df obtained an overall average improvement of 3.332$\%$ in all measures with a minimum of 2.196$\%$ and a maximum of 4.853$\%$ on F$_{\beta=1}$ and P@20, respectively. The 11-point result in the left figure of Figure 6.22 confirms the results in Table 6.25 in which the combination with df still perform slightly better than other combinations. From the values of the k parameter in Table 6.25, it is apparent that combining Equation 4.4 with df requires a smaller number of terms (the top-16 terms from each collection) to score its best performance while the combination with pf required 3.3 times more terms to achieve their best results. In the RRT experiments, the same integration with df achieved an improvement of 1.936$\%$ on the nDCG@4 measure compared to the integration with pf. While this improvement scored using the top-4 terms, the right figure of Figure 6.22 shows that the combination with df remains slightly better than other combinations for the first 25 terms measured by the nDCG metric.

In accordance with the same experiments conducted on the SIF model and reported in Table 6.23 and Figure 6.19, we can see that even in SIF2's experiments that the raw-statistics remains more representative compared to estimated ones (e.g., idf and $tfidf$). Both, pf and tf, continues to show competitive performance compared to the df. Moreover, integrating pf and tf with Equation 4.4 of the SIF2 models made them performed almost equally the same, as can be seen in Table 6.25 and Figure 6.22. Besides, the SIF2 equation also made both $tfidf$ and idf perform similarly, which was not the case in SIF's experiments. Overall, we can conclude that (1) taking the local statistical details of low-level terms into account and the (2) effective integration between them and the revised topical statistics and the df can estimate better weights that accurately represent the relevance of these terms to the user information needs, as demonstrated in the experiments.

- **Parameters Sensitivity**

  As an improved version of SIF, the SIF2 model inherited the same experimental parameters; V, which denotes the number of LDA topics; and k that represents the number of top topical terms discovered the model. The same experiments, which had been conducted for SIF

Table 6.25: The IF and RRT results of SIF2's main weighting function (Equation 4.4) integrated with different global statistics of low-level terms averaged over the 50 collections of the RCV1 dataset

	P@20	BP	MAP	$F_{\beta=1}$	IAP	nDCG@4	k
$sr_{D^+}(t) \cdot df(t)$	**0.605**	**0.504**	**0.535**	**0.491**	**0.557**	**0.472**	**16**
$sr_{D^+}(t) \cdot pf(t)$	0.577	0.493	0.517	0.480	0.536	0.463	53
$sr_{D^+}(t) \cdot tf(t)$	0.574	0.486	0.515	0.479	0.535	0.453	45
$sr_{D^+}(t) \cdot tfidf(t)$	0.444	0.382	0.395	0.403	0.423	0.124	39
$sr_{D^+}(t) \cdot idf(t)$	0.431	0.387	0.394	0.405	0.419	0.106	33
improvement%	+**4.853**%	+**2.208**%	+**3.548**%	+**2.196**%	+**3.855**%	+**1.936**%	

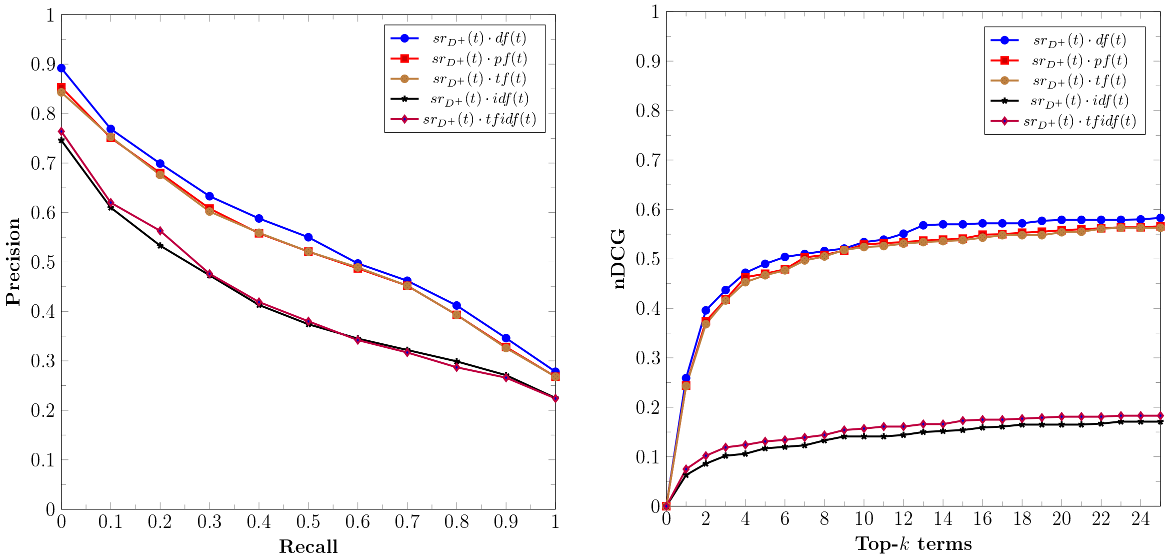


Figure 6.22: The results of the 11-point measure (left) and the results of the nDCG measure at top-25 terms (right) of SIF2's weighting function (Equation 4.4) with other global statistics averaged over the first 50 collections of the RCV1 dataset.

parameters sensitivity, were also repeated on SIF2 to verify how sensitive it is to these parameters. The results illustrated in Figures 6.23 and 6.24 show that SIF2 continues to inherit the insensitivity of the SIF model towards the two parameters. In the IF task, our SIF2 model has stable performance in all measures at any given value of the V parameter, as can be seen in the left figures of Figures 6.23 and 6.24, except some negligible fluctuations on the P@20 measure. The model also shows a stable performance after the top five topical terms ($k = 5$), as can be seen in the right figure of Figure 6.23, even though the best results reported in Tables 6.7 and 6.25 were for the top 16 words. However, and in the same figure, SIF2 performance in IF measured by the P@20 metric remains to show insignificant fluctuations. In the RRT task, our SIF2 model continues its insensitivity towards the V and k parameters as illustrated in the right figure of Figure 6.24.

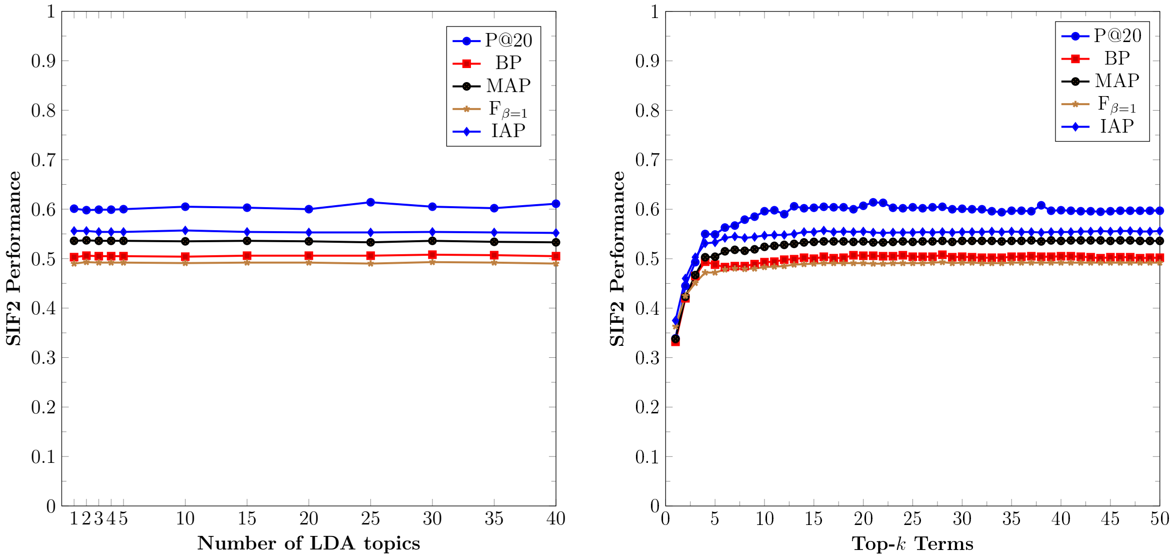


Figure 6.23: The SIF2 sensitivity to the number of LDA topics (left) and top-k terms (right) for the IF experiments.

6.9.3 The Proposed UR Method

- **UR Effects on Fusion Algorithms**

As observed from the extensive results presented in Section 6.8.3, the UR method effectively and significantly improved the performance of all the twelve different fusion-based TFS models in both the IF and RRT applications. The results experimentally demonstrated the merits of the UR method in which the uncertainties available in the positive feedback (i.e., relevant documents) can be reduced via the implicit estimation of the paragraph-relevance using latent topics. Inspired by the assumption of our SIF2 model in which a topical term has both local

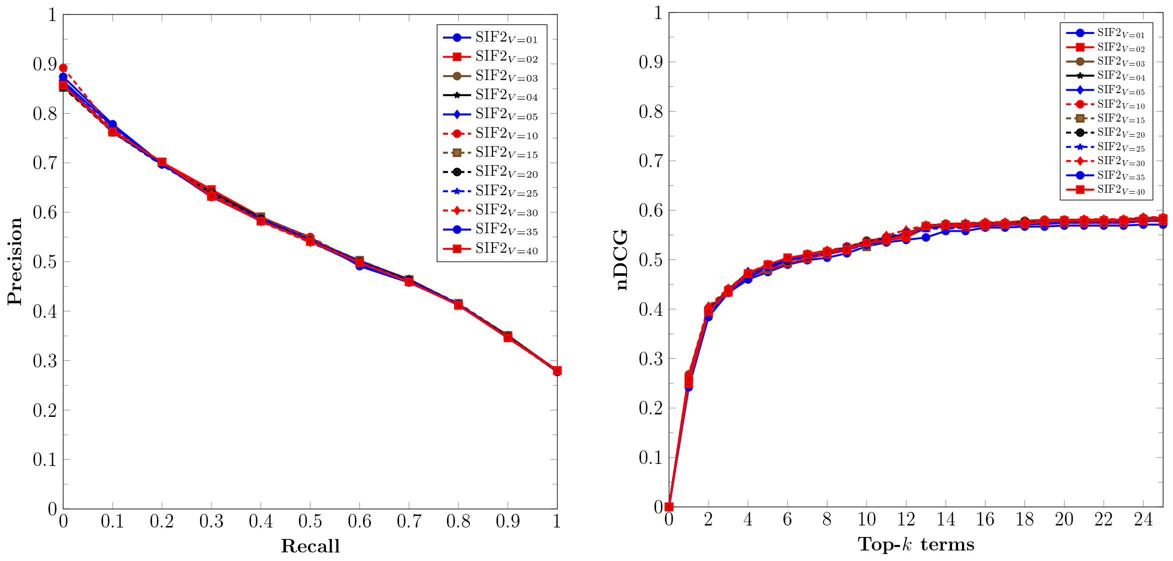


Figure 6.24: The SIF2 11-point results for IF (left) and SIF2 nDCG@k results for RRT (right) over different number of LDA topics.

and global significances, the UR method assumed that a paragraph has local relevance, at its document, as well as another global relevance at the entire collection of relevant documents. The fusion of the paragraph relevance scores indicates its significance to the topic(s) of interest in the collection that discusses user information needs. However, unlike SIF and SIF2 models, the UR method did not consider the terms-topics distributions because LDA estimates them from all terms in the collection paragraphs without paying attention to the evidence of relevance in these paragraphs knowing that some of these paragraphs can be irrelevant as illustrated in Figure 1.3. Instead, the UR method relied on raw frequency distributions of the terms in their documents and all paragraphs in the collection as these distributions show to be representative in revealing the importance of these terms as demonstrated in SIF and SIF2 experiments.

As observed from the results of the UR method, the amount of improvement in each feature set discovered by a specific TFS model varies depends on certain characteristics of the model. For example, for IF, the best performance and the highest improvements were achieved by the supervised early fusion models, especially the SVM, BM25 and Prob models, as it can be seen from Table 6.10 and Figure 6.14. We can speculate that (1) the effective use of negative feedback by these models; (2) the soundness of their weighting functions; and (3) the flexibility of the low-level terms discovered by these models have positively affected their performance. The UR method also brought the multi-topic representation to these term-based models. Also, from the feature fusion perspective, our UR method implicitly integrated topical and local statistical features with these models, which transferred them to hybrid fusion models (i.e., iSVM, iBM25 and iProb). However, while the unsupervised early fusion model (e.g., TFIDF) also gained significant improvement compared to their original performance, they did not show better performance than the supervised ones because they could not deal with negative documents and their weighting functions are not sufficient enough. Moreover, our UR method not only improved the performance of the early fusion models in IF. It also significantly improved their performance in the RRT task, as it can be seen in Table 6.11 and Figure 6.15. The example in Table 6.26 shows how the UR method managed to re-rank the original terms and bring forward the most relevant ones. As can be seen from the same table, the original SVM, BM25 and Prob models were not able to discover any of the human-identified relevant terms. However, by integrate them with our UR method, not only have the relevant terms started to appear among the top-10 terms, we argue that a more accurate weight is also assigned to the original terms as it confirmed by the models IF results.

An interesting observation is that our UR method effectively improved the performance of all unsupervised late fusion models in all experimental tasks. Integrating the UR method with the pattern-based PDS model not only significantly improved its original performance, but it also made it outperformed all the pattern-based topic models (i.e., PBTM-FP, PBTM-FCP, SPBTM and MPBTM) regardless of the type of patterns employed by these complex models. The UR method also not only brought the multitopic assumption to pattern mining, but it also provided an effective way to use patterns and alleviate the low-frequency of some specific patterns. The example in Table 6.26 illustrates the benefits that our UR method brought to the PDS model. It can be seen how the UR method re-ranked the PDS original terms and thus allows some specific terms that were appearing in low-frequent patterns to be highly ranked in the list, such as the term ’*espionage*’ and ’*secret*’. Our method also revised the original pattern-based term weight resulting in some scaling ups and downs of some terms. For example, the original PDS assigned a higher weight to the general word ’*economic*’ (0.238) while after the integration with the UR method the weight scaled down to (0.012) as general words are less specific. These benefits made the improved PDS (iPDS) achieved the best result (0.490) in RRT task measured by the nDCG@4 metric compared to all improved models.

More interestingly, the UR method also significantly improved the performance of LDA and

PLSA in both IF and RRT experimental tasks. Both LDA and PLSA do not distinguish the most relevant paragraphs even though LDA estimates the relevance of terms based on the topics extracted from all paragraphs in the collection, which improves the performance of LDA (LdaPara) compared to its performance using the whole documents (LdaDoc) as shown in Table 6.7. The new improvements made by integrating LDA and PLSA with the UR method can confirm (1) the effectiveness of the UR method in estimating the relevance of paragraphs and utilising them in reducing uncertainties in relevant documents; and (2) the existence of uncertainties in the terms-topics distributions knowing that our UR method does not use these statistical features because they might be affected by the uncertainties in some paragraphs. Overall, the UR method made several supervised and unsupervised performed comparably similar despite the differences in their algorithms, the feature they use or the fusion strategy they adhere to as illustrated in Figures 6.16 and 6.25.

Table 6.26: The top-10 stemmed terms from collection 101 of the RCV1 dataset, which about ’economic espionage’, discovered and ranked by different TFS models in which only iPDS, iSVM and iBM25 was able to select both of these relevant features.

iPDS		**PDS**		**iSVM**		**SVM**	
Term	*Weight*	*Term*	*Weight*	*Term*	*Weight*	*Term*	*Weight*
espionag	**0.896**	vw	0.617	secret	0.537	vw	0.419
secret	0.433	bill	0.343	compani	0.379	piech	0.239
crime	0.083	piech	0.340	**espionag**	**0.372**	men	0.218
compani	0.042	men	0.256	crime	0.292	bill	0.199
bill	0.036	**econom**	**0.238**	bill	0.234	photograph	0.175
econom	**0.012**	car	0.152	technolog	0.182	carmak	0.174
pass	0.006	photograph	0.150	**econom**	**0.181**	return	0.153
feder	0.006	carmak	0.150	pass	0.174	volkswagen	0.130
foreign	0.002	camera	0.134	foreign	0.146	gm	0.127
senat	0.001	volkswagen	0.125	piech	0.137	camera	0.125

iBM25		**BM25**		**iProb**		**Prob**	
Term	*Weight*	*Term*	*Weight*	*Term*	*Weight*	*Term*	*Weight*
bill	0.423	secret	0.130	bill	0.444	secret	0.126
secret	0.402	technolog	0.112	secret	0.428	crime	0.109
crime	0.302	crime	0.112	crime	0.324	pass	0.109
espionag	**0.264**	pass	0.112	compani	0.232	technolog	0.109
compani	0.237	fbi	0.098	pass	0.202	bill	0.094
pass	0.189	bill	0.098	technolog	0.202	cia	0.094
technolog	0.189	cia	0.098	theft	0.190	law	0.094
theft	0.181	law	0.098	feder	0.164	softwar	0.094
econom	**0.178**	softwar	0.098	**espionag**	**0.163**	fbi	0.094
feder	0.175	comput	0.098	law	0.158	comput	0.094

- **UR Effect on k Parameter**

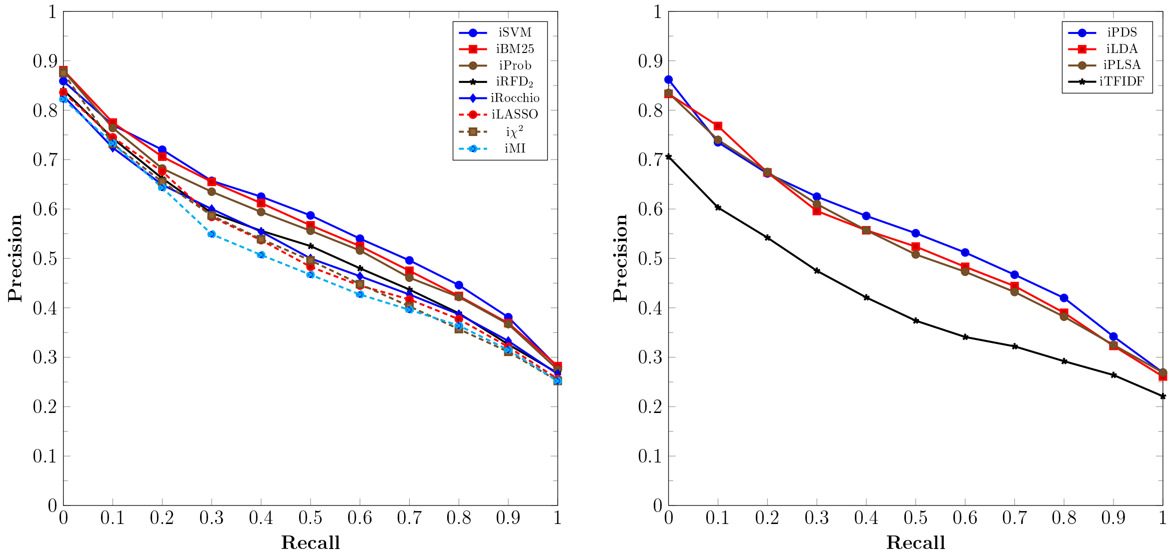


Figure 6.25: The 11-point results of supervised (left) and unsupervised (right) models after the integration with the UR method all averaged over the first 50 collections of the RCV1 dataset.

Figure 6.26 shows the best k value for each TFS model used in the UR experiments. Both values of k, for the original and improved model, are reported in the figure. It seems complicated to find any correlation between the use of the UR method and the k parameter because each model has its unique characteristics in dealing with the identification of relevant features. However, while applying the UR method significantly improved the performance of all models in both IF and RRT tasks, it also reduced the number of top terms (i.e., k value) needed to achieve the best performance for most models. Eight models out of twelve had their k values reduced after applying the UR method while the remaining four models got their k value increased. We speculate that the influence of the factors mentioned at the beginning of this section has made it difficult to establish any correlation between applying the UR method to any TFS model and the changing in the values of the k parameter.

6.9.4 The Proposed USIF Framework

The experimental results of our USIF framework presented in Section 6.8.4 clearly illustrated its superiority in discovering relevant features that represent user information preferences compared to the used baseline models. Unlike our SIF, SIF2 and UR models, the USIF framework employed multiple hybrid fusions of different lexical and statistical features that were extracted from a collection of relevant documents using document clustering and topic modelling algorithms and the global statistics of the collection. The framework utilised the hybrid fusions to select and then re-weight topical terms that appear in equally relevant but unbalanced clusters.

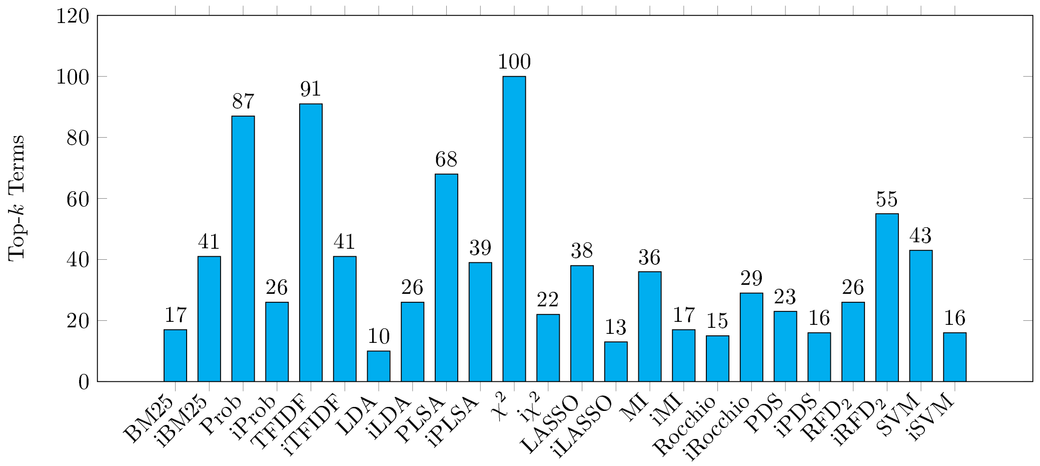


Figure 6.26: The best k value for each TFS model after and before applying the UR method.

A conceptual agglomeration technique is developed to select a specified set of intra- and inter-cluster terms based on a score fusion scheme ($r(t_i)$). Then, the relevance of these representative terms is estimated based on their topical and thematic significances as well as their document frequencies in the collection. More analysis of the proposed USIF framework is given below.

- **Feature Selection Versus Feature Weighting**

  Generally, the proposed USIF framework has dealt with feature selection and feature weighting as two different problems. The representativeness of the selected topical terms (i.e., feature selection) and their relevance estimated jointly from topical significance and the thematic significance (i.e., feature weighting) have substantial contributions to the performance of the proposed USIF framework. To analyse these contributions, we have designed seven scenarios. The scenarios (scen-1 to scen-7) are summarised in Table 6.27. Each scenario is designed to analyse the effect of a change in one or more components of the proposed framework on its overall performance. The corresponding experimental results using the seven performance measures (i.e., P@20, BP, MAP, F$_{\beta=1}$, IAP, 11-point and nDCG@k) are shown in Table 6.28 and the left figure of Figure 6.27. The key observations obtained from these scenarios can be summarised as follows:

  (**a**) The performance of scen-1 is better than scen-2 and scen-3. Scen-1 uses topical significance $w_z(t_i)$ as $r(t_i)$, while scen-2 uses term frequency $tf(t_i)$ as $r(t_i)$ and scen-3 uses term frequency-inverse document frequency $tfidf(t_i)$ as $r(t_i)$, all learned from the corresponding clusters. This means $w_z(t_i)$ is better in revealing the representativeness of intra-cluster terms than simply using $tf(t_i)$ and $tfidf(t_i)$ as an estimation of $r(t_i)$. Further, from the results of scen-2 and scen-3, we can see that integrating our conceptual agglomeration of intra- and inter-cluster terms with the informativeness of our topical and thematic significances as well as the document frequency greatly improved the original performance of both $tf(t_i)$ and $tfidf(t_i)$.

(**b**) The performance of scen-1 is significantly better than scen-4. To select representative terms, scen-1 uses conceptual agglomerate of clusters' topical terms and the $r(t_i) = w_z(t_i)$, while scen-4 uses only the $r(t_i) = w_z(t_i)$. This means conceptual agglomeration of clusters' terms has a significant contribution to the performance of the USIF framework. However, while the selected terms are different in each of these scenarios, both of them use the same relevance score fusion function (i.e., $w(t_i)$), which made scen-4 achieve competitive results.

(**c**) The performance of scen-1 is marginally better than scen-5 even though both scenarios share the same selected set of topical terms. The only difference is the absence of using global statistics represented in our framework by the document frequency $df(t_i)$. This means that $df(t_i)$ has a marginal contribution in estimating the relevance of topical terms based on the BP and F$_{\beta=1}$ measures.

(**d**) Performances of scen-6 and scen-7 are significant and similar as they use the same set of topical terms selected by our conceptual agglomeration technique. As the estimation of informativeness, scen-6 uses topical significance $w_z(t_i)$, while scen-7 uses thematic significance $w_g(t_i)$. This means thematic significance $w_g(t_i)$ is as essential as topical significance $w_z(t_i)$ for estimating the relevance of the selected topical terms, especially when both significances integrated together.

(**e**) The performance of scen-5 is significantly better than scen-6 and scen-7. As the estimation of the relevance, scen-5 jointly uses the topical significance $w_z(t_i)$ and the thematic significance $w_g(t_i)$, while scen-6 uses only the topical significance $w_z(t_i)$ and scen-7 uses only thematic significance $w_g(t_i)$. This means the relevance of topical terms should be estimated jointly from both topical significance and thematic significance.

Overall, the previous scenarios demonstrated the importance of each component of our USIF framework and how they performed when they integrated to select and then re-score relevant topical terms that describe user information needs. Most importantly, the scenarios illustrated the fact that term selection and term weighting can differ from each other, and an effective integration between them can result in significant performance for unsupervised relevance discovery.

Table 6.27: A set of different scenarios designed for analysing the fusion hypothesis of the USIF framework

Scen-1	Our USIF (use conceptual agglomeration of clusters and $r(t_i) = w_z(t_i)$ learned from each cluster to select a set of representative topical terms, and use $w(t_i) = w_z(t_i) \times w_g(t_i) \times df(t_i)$ to weight these terms).
Scen-2	Use $r(t_i) = tf(t_i)$, term frequency learned from each cluster.
Scen-3	Use $r(t_i) = tfidf(t_i)$, term frequency-inverse document frequency learned from each cluster.
Scen-4	Instead of using clustering, use $w_z(t_i)$ learned from the whole document collection to select a set of representative topical terms.
Scen-5	Use $w(t_i) = w_z(t_i) \times w_g(t_i)$ to weight topical terms.
Scen-6	Use $w(t_i) = w_z(t_i)$ to weight topical terms.
Scen-7	Use $w(t_i) = w_g(t_i)$ to weight topical terms.

Table 6.28: The results of the scenarios in Table 6.27 for IF and RRT tasks using all measures averaged over the first 50 document collections of the RCV1 dataset

Scenario	P@20	BP	MAP	F$_{\beta=1}$	IAP	nDCG@4
scen-1	**0.616**	**0.518**	**0.550**	**0.500**	**0.571**	**0.502**
scen-2	0.570	0.495	0.517	0.482	0.539	0.460
scen-3	0.570	0.483	0.507	0.477	0.529	0.447
scen-4	0.576	0.487	0.514	0.481	0.534	0.457
scen-5	0.584	0.500	0.523	0.486	0.544	**0.502**
scen-6	0.550	0.467	0.488	0.468	0.510	**0.502**
scen-7	0.555	0.471	0.496	0.473	0.517	**0.502**

- **Parameters Sensitivity Test**

  The proposed USIF framework has three parameters: the number of document clusters (L), the number of LDA topics (V) and the number of top representative topical terms (k). Because it is challenging to decide the optimal value of L for a given document collection [Das et al., 2008, Jain, 2010, Liu and Croft, 2004], a trial and error approach was used to develop the line equation presented in Section 6.7 to predetermine the value of L. Regarding the number of topics V, it is expected that USIF would not be sensitive to this hyperparameter because the topical and thematic significances of terms in the framework are estimated by our SIF model and the adapted version of our UR method, which already proven to be insensitive to V. Moreover, given a topical term t_i, the $w(t_i)$ of that term is estimated using all relevant topics in the D^+ collection and not based on any specific topic z_j. Thus, regardless of the number of topics V generated from D^+, they all represent the same collection and $w(t_i)$ should not strongly depend on their numbers, which is denoted by V. The results of the sensitivity test for USIF over different numbers of topics are given in Figure 6.28. The results confirm our expectation and show that $w(t_i)$ is quite insensitive to the V parameter.

  It is also expected that the performance of our USIF framework is stable for a range of top topical terms (i.e., k) because USIF was developed to treat term selection and term weighting as two independent stages in the framework. Thus, if some nonrepresentative terms are accidentally selected by the first stage due to the nonoptimal number of clusters estimated from the collection, then, the second stage should be robust enough to weight them as much less important compared with the most representative terms in the collection. The performance sensitivity of USIF for a range of top-k terms (from $k = 1$ to $k = 150$) is given in the right figure of Figure 6.27. It shows that after the 20 top terms, the performance becomes stable with occasional small fluctuations, which supports our expectation.

6.9.5 The Proposed SSIF Framework

The results presented in Section 6.8.5 show the performance superiority of our SSIF framework compared to all fusion-based TFS baseline models of relevance discovery. The sophistication of SSIF and, more specifically, the effective use of negative feedback collections largely contribute to its outstanding performance. As in our USIF framework, SSIF also deals with feature selection and feature weighting as two independent problems through the integration of multiple hybrid fusion-based models. However, and unlike USIF, our SSIF framework utilises

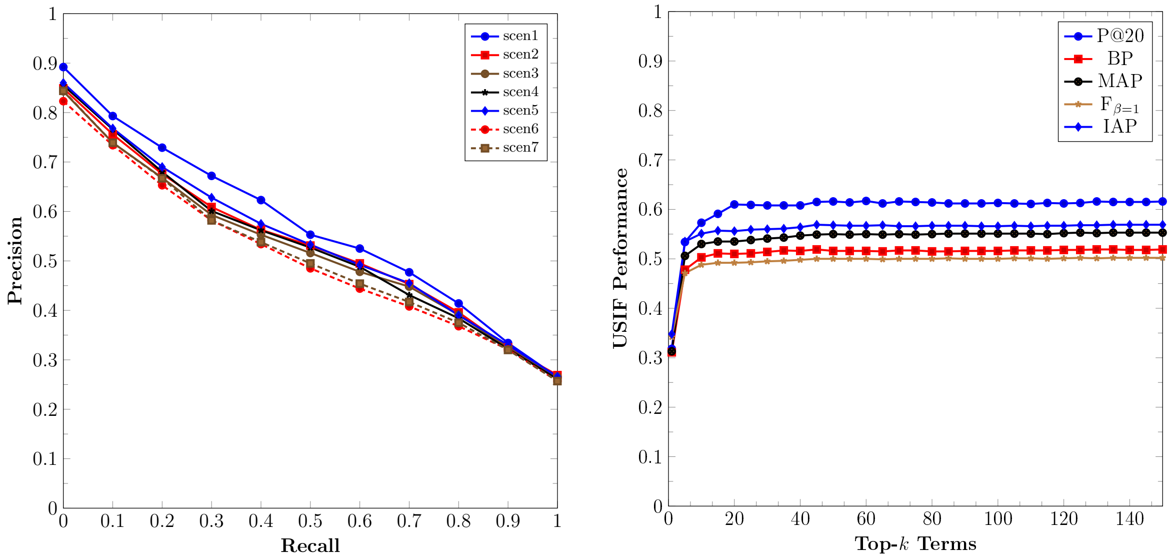


Figure 6.27: The 11-point result of the scenarios in Table 6.27 (left) and the results of USIF sensitivity test to the k parameter (right) all for IF and averaged over the same RCV1 collections.

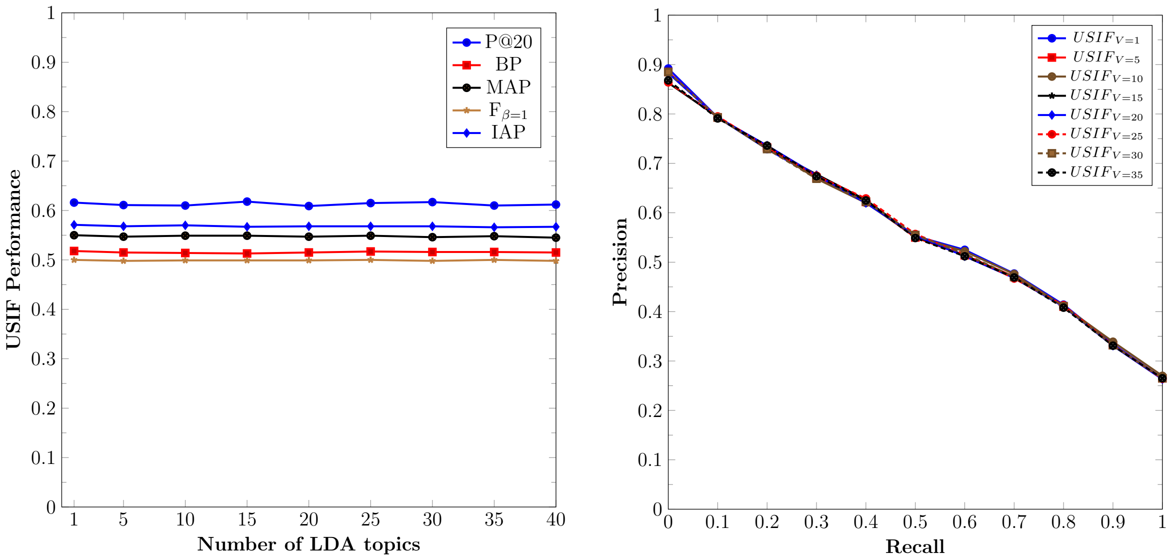


Figure 6.28: The IF results of USIF sensitivity test to the V parameter using all measures averaged over the first 50 collections of the RCV1 dataset.

supervised learning algorithms to select some discriminatively specific features and re-weight them using unsupervised learning algorithms. In the light of SSIF experimental results, we discuss SSIF's hypothesis in which supervised feature selection can discover more specifically relevant features, but unsupervised feature weighting can better estimate their informativeness.

As illustrated in Figure 5.5, the multiple hybrid fusions of different lexical and statistical features extracted from the positive and negative feedback using supervised and unsupervised algorithms have made our SSIF framework significantly outperforms all state-of-the-art baseline models in discovering relevant features that describe user information needs. The SSIF framework was developed on the basis that the discovered features set must be (1) specific to the main topics of interest in the document collection. SSIF effectively employed the integration of our UR method, BM25 and SVM algorithms to meet this criterion. Also, as there might be several topics and themes in the collection, the relevance of this set of features must be (2) informative about the essential aspects of meanings of these topics and themes. To meet this condition, our SSIF framework adopted both the topical and thematic significances in a similar way as in the USIF framework. Further, the feature set must be (3) globally representative to the given collection not to a larger document. Thus, and to meet this condition, the SSIF framework used the global statistics represented by the document frequency in this case.

From Table 6.19 and Figure 6.18, we can see that the sophistication of our SSIF framework has made it significantly outperformed all supervised baseline models. Compared to the best model in the group, the RFD_2, it is clear that SSIF effectively selected more specifically relevant features compared to the RFD_2 through the integration of the UR, BM25 and SVM models. SSIF also estimated more accurately informative scores to these features via the joint probability of topical and thematic significances of these features combined with their document frequencies. Thus, it is apparent that the integration between an effective supervised selection and unsupervised weighting of features can significantly discover relevant features that represent user information needs. Regarding the RFD_2, we can speculate that the (1) absence of multi-topic assumption; (2) the challenge of selecting representative patterns from both positive and negative feedback and (3) ignoring the available uncertainties in positive documents have contributed to its inferior performance compared to our SSIF framework. Moreover, despite the cleverness of the RFD_2 specificity function in classifying features to general, specifically positive and specifically negative, we argue that this function is sensitive to the type of pattern in use, the size of terms space and most importantly to the experimental coefficients. However, and based on the experimental results in Figures 6.28 and 6.14, we can see that using the topical and thematic significances as well as applying the UR method to SVM can make the performance of discovering relevant features robust and insensitive to any parameters.

The MPBTM model is one of the state-of-the-art baseline models in discovering relevant features that discuss user information needs thru the integration of patterns and topics. It is the best among all unsupervised baseline models, as shown in Table 6.19. However, our SSIF framework significantly outperformed MPBTM in all performance measures. The effectiveness of SSIF in utilising the negative documents has given it the superiority over MPBTM. Besides, the MPBTM model effectively exploited the semantics of both patterns and topics to rank specifically relevant documents that meet user information interests. Nevertheless, the model failed to address the uncertainties in training documents as it assumed all documents contents are important, which resulted in either selecting irrelevant features or inaccurately estimating relevant documents. Also, it can be argued that selecting some patterns and ignoring others can cause the loss of some important features, especially the less frequent ones. Moreover, the MPBTM model seems to be sensitive to the number of latent topics (i.e., the V parameter) as its performance significantly fluctuated with different V. However, our SSIF was more stable, robust and insensitive to all its experimental parameters.

6.9.6 Comparison of Proposed Techniques

In the previous sections, we presented, analysed and discussed some of the experimental results of our proposed models and frameworks. We also provided detailed comparisons between them and many popular and state-of-the-art baseline models. In the following sections, we compare and briefly discuss the performances of SIF, SIF2, USIF and SSIF in IF and RRT as illustrated in Table 6.29 and Figure 6.29.

- **SIF2 Versus SIF**

  The first part of Table 6.29 shows the comparison between the performances of SIF2 and SIF in both IF and RRT experimental tasks. SIF2 performance was consistently better than SIF by an average improvement of 5.434$\%$ in all measures. SIF2 achieved its best performance (6.980$\%$) on the MAP metric for the IF task, which is considered the most important measure in IR and IF experiments. However, SIF2 minimal performance compared to SIF was in RRT by an average improvement of 3.282$\%$ on the nDCG@4 metric. The 11-point results in Figure 6.29 (left) illustrates that SIF2 was performing better than SIF, especially in the last nine recall levels. However, in the RRT task for the first 25 terms, SIF2 was slightly better than SIF, as shown in Figure 6.29 (right). Overall, all these results confirm the validity of SIF2's assumption that a topical term should not be equally relevant in every document of the collection. Our SIF2 model shows that the accurate revision of the global relevance details of features can alleviate the uncertainties available in the entire collection to a considerable extent. The model demonstrates that localising global relevance details of topical terms can estimate more accurate weights to these terms and thus resulting in discovering more specifically relevant terms, especially when they are integrated with informative global statistics.

Table 6.29: A comparison between the performances of all proposed models and frameworks in IF and RRT tasks using six evaluation measures averaged over the first 50 collections of the RCV1 dataset

Model	P@20	BP	MAP	$F_{\beta=1}$	IAP	nDCG@4
SIF2	**0.605**	**0.504**	**0.535**	**0.491**	**0.557**	**0.472**
SIF	0.567	0.475	0.500	0.473	0.527	0.457
improvement %	+**6.702**%	+**6.133**%	+**6.980**%	+**3.795**%	+**5.709**%	+**3.282**%
USIF	**0.616**	**0.518**	**0.550**	**0.500**	**0.571**	**0.502**
SIF	0.567	0.475	0.500	0.473	0.527	0.457
improvement %	+**8.642**%	+**8.966**%	+**9.825**%	+**5.814**%	+**8.312**%	+**9.847**%
SSIF	**0.631**	**0.550**	**0.576**	**0.515**	**0.592**	0.420
SIF	0.567	0.475	0.500	0.473	0.527	**0.457**
improvement %	+**11.287**%	+**15.831**%	+**15.149**%	+**8.809**%	+**12.354**%	−**8.096**%
USIF	**0.616**	**0.518**	**0.550**	**0.500**	**0.571**	**0.502**
SIF2	0.605	0.504	0.535	0.491	0.557	0.472
improvement %	+**1.818**%	+**2.669**%	+**2.660**%	+**1.945**%	+**2.462**%	+**6.356**%
SSIF	**0.631**	**0.550**	**0.576**	**0.515**	**0.592**	0.420
SIF2	0.605	0.504	0.535	0.491	0.557	**0.472**
improvement %	+**4.298**%	+**9.137**%	+**7.636**%	+**4.831**%	+**6.286**%	−**11.017**%
SSIF	**0.631**	**0.550**	**0.576**	**0.515**	**0.592**	0.420
USIF	0.616	0.518	0.550	0.500	0.571	**0.502**
improvement %	+**2.435**%	+**6.300**%	+**4.847**%	+**2.831**%	+**3.732**%	−**16.335**%

- **USIF Versus SIF**

  The performance comparison between the SIF model and the USIF framework in IF and RRT experiments is given in the second part of Table 6.29. As can be seen, USIF significantly outperformed SIF in all measures for both experiments by an overall average improvement of 8.568%. In IF, USIF achieved its minimum improvement against SIF on the $F_{\beta=1}$ metric by an average of 5.814%, and its maximum improvement for the same task was 9.825% on the MAP measure. Figure 6.29 (left) confirms the superiority of USIF in IF as it achieved higher

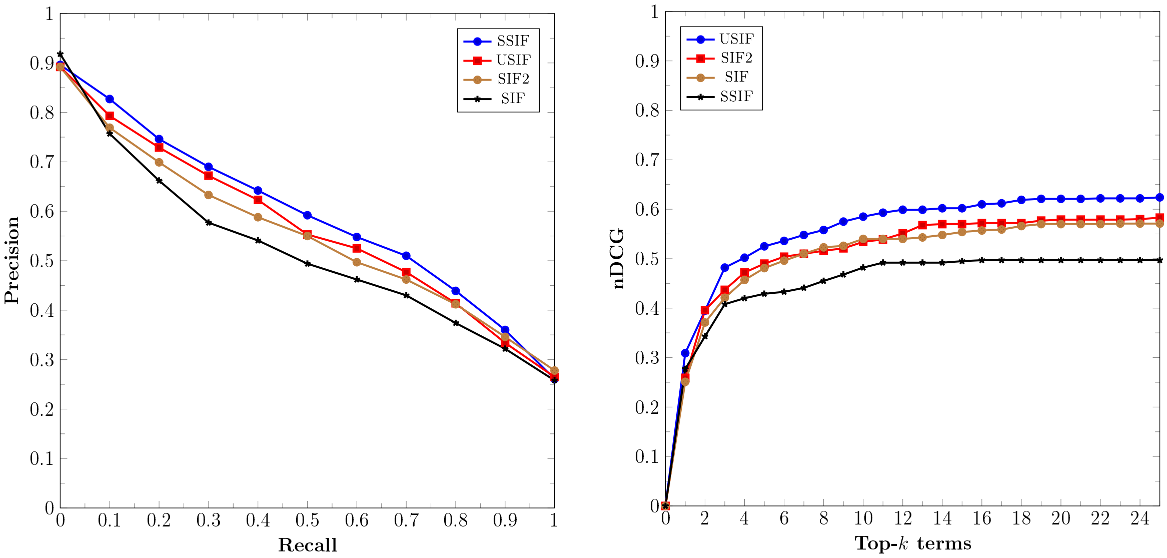


Figure 6.29: The 11-point (left) and nDCG@k (right) results for IF and RRT, respectively, for all the proposed models and frameworks averaged over the first 50 collections of the RCV1 dataset.

average precision scores at most recall levels compared to SIF. In RRT, USIF performance was significantly better than SIF by an average improvement of 9.847$\%$ on the nDCG@4 measure, as shown in Table 6.29. Figure 6.29 (right) also shows that USIF is performing consistently better in RRT at any k value compared to the SIF model. All these significant improvements of USIF over SIF come as a result of the sophistication of the USIF framework in integrating topic modelling, document clustering and global statistics to discover representative features and estimate their informativeness as previously demonstrated in Section 6.9.4.

- **SSIF Versus SIF**

The third part of Table 6.29 presents the results of the supervised SSIF framework and the unsupervised SIF model. From the improvement$\%$ row of that part of the table, we can see than SSIF significantly outperformed SIF in IF in all five measures. SSIF maintained an overall average improvement of 12.686$\%$ over SIF performance with a maximum of 15.831$\%$ and a minimum of 8.809$\%$ on the BP and $F_{\beta=1}$ measures, respectively. The 11-point results in Figure 6.29 (left) supports the previous measures and shows its superiority over the SIF model at nearly all recall levels. However, in RRT, SSIF underperformed compared to SIF with an average of -8.096% on the nDCG@4 metric, as illustrated in Table 6.29. This can be seen clearly in Figure 6.29 (right) in which SIF performed much better than SSIF, especially in the last 24 values of the k parameter. In general, the superiority of SSIF over SIF comes as a result of its sophistication in selecting and weighting specifically relevant features through the integration of supervised and unsupervised learning algorithms. However, USIF inferior performance in RRT compared to SIF is expected as most of the supervised models used in the experiments of this thesis did not perform well compared to their unsupervised counterparts. We can speculate that the reason behind the poor performance of supervised models in RRT is that the human-identified relevant words are not comprehensive and only focus on general ones.

- **USIF Versus SIF2**

  The results of USIF and SIF2 for IF and RRT experiments are presented in the fourth part of Table 6.29. Both USIF and SIF2 are unsupervised TFS methods and performed competitively in our experimental tasks. However, USIF performed better than SIF2 in IF by an overall average improvement of 2.311$\%$. It achieved a minimum improvement of 1.818$\%$ on P@20 and a maximum of 2.669$\%$ on BP. On the 11-point measure, both techniques competed with each other, but USIF scored higher precision than SIF2 in several recall levels, as illustrated in Figure 6.29 (left). In RRT, USIF significantly outperformed SIF2 by an average improvement of 6.356$\%$ on the nDCG@4 measure, as shown in Table 6.29. Figure 6.29 (right) shows the performance of USIF and SIF2 in RRT for the top-25 words in which USIF maintained greater improvements at all terms. Despite the sophistication of the USIF framework, the SIF2 model demonstrated an adequate competency compared to it, especially in IF. However, USIF illustrated its capability in selecting representative features as can be seen in its RRT results. Further, while USIF was developed before SIF2 in this thesis, it might be feasible to employ SIF2 capabilities in a similar research objective as USIF's.

- **SSIF Versus SIF2**

  The fifth part of Table 6.29 shows the results of SSIF and SIF2 for both IF and RRT tasks. As can be seen from that part of the table, SSIF outperformed SIF2 in IF and achieved a minimum average improvement of 4.298$\%$ and a maximum of 9.137$\%$ on the P@20 and BP measures, respectively. Overall, SSIF maintained better performance than SIF2 by an average improvement of 6.438$\%$ in all measures. The 11-point results in Figure 6.29 (left) confirmed SSIF better performance over SIF2 in IF. However, on the contrary, SIF2 significantly outperformed SSIF in the RRT task with an average improvement of 11.017$\%$ on nDCG@4 metric, as shown in Table 6.29. Moreover, SIF2 was performing significantly better than SSIF nearly at any given top-k keyword, as illustrated in Figure 6.29 (right). As a supervised framework, SSIF improvements over SIF2 were expected because SIF2 is considered as an improvement to the SIF model. It is apparent that the integration of different supervised and unsupervised algorithms made SSIF capable of selecting and weighting specifically relevant features compared to the unsupervised SIF2. The poor performance of SSIF in RRT can be justified as in the case of SSIF versus SIF mentioned previously.

- **SSIF Versus USIF**

  A comparison between the performances of our supervised SSIF and unsupervised USIF frameworks are given in the last part of Table 6.29. In IF, SSIF maintained better performance than USIF by an overall average improvement of 4.029% across all five measures. SSIF performed minimally by achieving an average improvement of 2.435% on the P@20 metric compared to USIF. Its maximum performance in IF was measured by the BP metric and obtained an average improvement of 6.300% over USIF. The 11-point measure in Figure 6.29 (left) confirmed the effectiveness of SSIF in IF against USIF as it achieved higher precision scores at nearly all the 11 recall levels. However, for the RRT and as shown in Table 6.29, USIF was superior in performance than SSIF and outperformed it significantly by an average improvement of 16.335% on nDCG@4. Figure 6.29 (right) clearly shows USIF superiority over SSIF in RRT as it maintained a significant performance at each top-k term for the first 25 terms. It is apparent that the use of negative documents has made SSIF better than USIF, especially in selecting a set of specifically relevant features. It also alleviates the problem of general features that keep appearing in both positive and negative training documents. While the two frameworks deal with the problems of feature selection and feature weighting differently, they both demonstrated that the accurate integration of different lexical and statistical features extracted by supervised and/or unsupervised techniques could discover more representative features that describe user information needs and thus achieve higher performance.

6.10 Chapter Summary

In this chapter, the extensive experiments conducted to evaluate the proposed fusion-based TFS models and frameworks were reported. The evaluation hypotheses and the experimental design were also described. The standard experimental benchmark that includes the RCV1 dataset and the TREC-11 topics for IF and seven popular performance measures were presented in the chapter including the statistical significance test; the Student's Paired T-Test and Percentage Change. Many different state-of-the-art baseline models were also briefly described and used to evaluate the proposed methods. The experimental results were reported in different forms and compared to the baseline results to show the superiority of the proposed models and frameworks in selecting and weighting relevant features. The nDCG@k measure clearly showed that SIF, SIF2, UR, USIF and SSIF were able to discover relevant features that match those identified by domain experts. The results were also discussed and analysed using many scenarios to demonstrate the robustness and effectiveness of fusion-based techniques and the proposed solutions for the discovered problems of selecting relevant features under uncertainties as well as those of the topic modelling algorithms. The next chapter concludes this thesis and describes its contributions. It also discusses the identified limitations and some possible future directions for the research presented in this thesis.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

For more than a decade, topic modelling has been extensively used in TM to enhance the automatic discovery of knowledge from texts in the form of latent topics. LDA is the most widely used probabilistic topic modelling algorithm, superseding its predecessor, the PLSA. Both techniques have been adapted substantially to suit multiple applications. Many existing projects focus on improving the algorithms' efficiency, scalability and quality of generated latent topics. However, using these topics to identify relevant features from a collection of documents that describes user information needs is ineffective for several reasons. First, LDA cannot generalise the weight of topical terms that appear across different entities in the collection. Second, LDA favours the most frequently discussed subjects in the collection, which can overshadow less frequent but equally important subjects. Third, LDA does not provide a mechanism to consider the hierarchical topical features of documents and the skewness of terms distribution across them when estimating the weight of topical terms. Further, LDA cannot deal with uncertainties in relevant features, as it does not consider passage level evidence. Finally, LDA cannot discover relevant features using both positive and negative documents.

Data fusion is a well-known approach that is proven to be effective in estimating relevant information by combining different features that represent various aspects of the data. In this thesis, effective fusion-based models and frameworks for relevant text feature weighting and selection have been proposed. The models and frameworks are developed to overcome the already noted challenges of topic modelling and have been integrated with both supervised and unsupervised learning algorithms and global statistics for better performance. A new and elegant ERS theory was developed to efficiently and effectively model the complex relationships between different entities in document collection and to manage the different types of fusion between their features. Utilising the proposed models to improve the performance of existing relevant feature discovery techniques was also investigated.

This thesis presents research in the field of TFS for relevance discovery based on the concept of data fusion. Different fusion strategies have been adopted and integrated to combine latent topical features with global statistics, as well as supervised and unsupervised learning algorithms. The SIF model (Chapter 3) was developed based on the concept of hybrid fusion to discover relevant topical terms by generalising their weight to the collection level. The SIF2 model (Chapter 4) re-visits the concept of generalised term weight in SIF and is introduced to integrate late and early fusion strategies to relax the weight generalisation assumption. The UR method (Chapter 4) was developed to reduce uncertainty in relevant features discovered by existing models. USIF is a TFS framework built around the concept of multiple hybrid fusions to integrate topic modelling, document clustering and global statistics for better relevant feature discovery (Chapter 5). SSIF is another framework introduced in Chapter 5 and developed to discover relevant features from both relevant and irrelevant documents. Within each model and framework, various mechanisms are proposed including the ERS theory, term weighting schemes, term scaling functions, concept agglomeration, topical significance and thematic significance to accomplish the aims of the proposed models and frameworks.

The proposed TFS models and frameworks were experimentally evaluated (Chapter 6) for IF and RRT using the 50 expert-assessed collections from the standard RCV1 dataset, their TREC relevance judgements and seven widely adopted performance measures. The results show that the proposed models and frameworks significantly outperform all state-of-the-art baseline models regardless of the text feature or fusion strategy used.

In the following, Section 7.2 presents the main contributions of this research and Section 7.3 discusses the limitations of the study and recommends future work in feature selection and weighting for relevance discovery.

7.2 Contributions

This thesis makes several contributions to the field of relevant feature discovery under uncertainty using fusion-based approaches.

- **Solving topic modelling problems**: It is possible to generate a specific number of latent topics from a document collection using probabilistic topic modelling algorithms. These topics have been used extensively in a range of TM applications. However, utilising these topics in TFS for relevance discovery is ineffective due to the specific characteristics of generating algorithms (see Section 7.1). In this thesis, effective models and frameworks have been proposed to circumvent the limitations of topic modelling by adopting and integrating fusion strategies with global statistics and learning algorithms. An innovative ERS theory was developed to model the proposed fusion strategies. Further, effective weighting and scaling formulas were introduced to weigh or re-rank relevant features so these features can be used in TM systems.

- **Innovative Hybrid Fusion-Based TFS model**: An effective TFS model, SIF, was developed based on a hybrid fusion strategy to discover relevant features (i.e., topical terms). The model implements three knowledge discovery steps, including (1) generating latent topics, (2) modelling hybrid fusion and (3) ranking topical terms. In the first step, SIF uses the LDA to generate useful topics from all paragraphs in a collection of relevant documents. The topics reduce the dimensionality of the collection and adequately represent useful information (e.g., subjects or themes) discussed in the relevant paragraphs. In the second step, multiple random sets are extended to manage the hybrid fusion strategy of different features between three entities in the collection; namely, paragraphs, topics and terms. This is achieved by modelling the complex relationships between these entities with a probability function measuring the strength of each relationship. In the final step, an effective global term weighting scheme is introduced based on the ERS to rank topical terms (i.e., relevant features). To the best of our knowledge, SIF is the first hybrid fusion model that uses multiple ERSs for TFS. The SIF model was extensively tested for IF and RRT and showed significant performance compared to many competent baseline models of relevance discovery. A full description of the proposed SIF model can be found in Chapter 3 and a detailed experimental evaluation is presented in Sections 6.8.1 and 6.9.1.

- **Effective Hierarchical Feature Fusion TFS model**: A new and highly effective TFS model for relevance discovery, SIF2, was developed based on the integration of early and late fusion strategies of hierarchical features. Unlike SIF, which assumes that identical topical terms are equally important in each relevant document, SIF2 relaxes this assumption and differentiates these terms based on their local statistics in each document. The SIF2 model adopts the same knowledge discovery steps as SIF, but it differs in the last two steps and introduces a new global weighting function. First, in the fusion modelling step, an extra ERS is introduced to model the relationship between a fourth entity; namely, the document and its paragraphs. Further, the function that represents the relationship between a term and latent topics is updated to allow topical terms to be deployed based on their distributions in each document. In the final step, the term weighting scheme is also updated to reflect the changes in the second step. The global weighting function can assign a more representative fused score to topical terms, expressing the integration between late and early fusion of features. The proposed SIF2 model was extensively evaluated and the experimental results demonstrate its significant performance and confirm its merits. Chapter 4 describes SIF2 in detail and Sections 6.8.2 and 6.9.2 discuss its experimental evaluation.

- **Innovative Uncertainty Reduction Method**: Another effective late fusion-based technique, the UR method, is proposed to reduce uncertainties in relevant features discovered by various existing TFS models. The uncertainties are introduced when these models consider the entire contents of a document knowing that a document can be labelled as relevant even if it has a small part(s) that matches what the user prefers. The UR method adheres to the same knowledge discovery steps as the SIF2 model. However, the newly developed ERS in the UR method does not consider the topic–term relationship, because LDA estimates the term–topic distribution using all the content of documents or paragraphs in the collection without distinguishing relevant passages. Instead, the ERS models the term–paragraph relationship. A new term weight scaling function was developed and used to re-rank relevant features discovered by different TFS techniques. To the best of our knowledge, the proposed UR method is the first of its kind that estimates the passage level relevance without an explicit query (i.e., a search guide) and uses multiple ERSs to model the hybrid fusion of different features from a document collection. The proposed UR method was tested extensively using many existing relevance discovery models. The experimental results show that the proposed method significantly improved the performance of these models for relevant feature selection. The UR method is described in more detail in Chapter 4 and its experimental evaluation is fully reported in Sections 6.8.3 and 6.9.3.

- **Novel Unsupervised TFS Framework**: A highly effective two-stage TFS framework, USIF, was developed based on the integration of multiple early and late fusions for relevant feature discovery. The framework treats term selection and weighting as two independent processes. First, the integration of document clustering and topic modelling was developed, in which the same knowledge discovery steps of SIF were used with every cluster of relevant documents. Then, a concept agglomeration technique was proposed to discover representative terms among the many intra- and inter-cluster topical terms. Further, an effective line-fitting equation was developed to pre-select the number of clusters. Second, an effective collection level feature-weighting technique was used based on the linear combination between the SIF model and a modified UR method. The SIF model was used to estimate the topical significance of topical terms in the collection, while the modified UR method was adopted to emphasise terms appearing in more relevant passages (i.e., paragraphs). To the best of our knowledge, USIF is the first TFS framework that integrates multiple early and late fusions of different features discovered by unsupervised methods; namely, document clustering, topic modelling and global statistics. Such sophisticated fusions are elegantly modelled by the multiple ERSs. The framework was extensively tested and the experimental results show its significant performance compared to many supervised and unsupervised baseline models. Chapter 5 discusses the details of USIF and the results of its evaluation are reported, analysed and discussed in Sections 6.8.4 and 6.9.4.

- **Effective Supervised TFS Framework**: Another highly effective TFS framework was introduced to discover relevant features not only from the positive (i.e., relevant) documents, as in SIF, SIF2, UR and USIF, but also from the negative (i.e., irrelevant) documents. Thus, the framework is fully supervised and is referred to as SSIF. As with USIF, SSIF uses multiple ERSs to model the integration of early and late fusion of informative features. Also, SSIF treats feature selection and feature weighting as two independent tasks in two distinct stages. First, specific features are selected using a supervised algorithm (e.g., SVM) after integration with the UR method. Second, informative features are learned and weighted in an unsupervised way using the combination of SIF, the UR method and global statistics. Finally, an efficient tactic is introduced to combine the output of the two stages. The proposed SSIF framework provides an effective method for discovering relevant features from both positive and negative documents by combining both supervised (i.e., SVM) and unsupervised (i.e., LDA) algorithms. The SSIF framework is innovative when dealing with the challenging problems of topical terms that frequently appear in both positive and negative contexts. The experimental results confirm the superiority of SSIF compared to major supervised and unsupervised baseline models. The proposed SSIF framework is described in detail in Chapter 5, while its experimental evaluation is discussed and analysed in Sections 6.8.5 and 6.9.5.

In summary, the research presented in this thesis demonstrates the adoption of different fusion-based techniques in TFS for relevance discovery.

1. The SIF model adopts a hybrid fusion strategy to select informative features at the collection level, which is achieved by:
   - generating latent topics from all paragraphs in the collection
   - extending multiple random sets to model the complex relationships between different entities in the collection from which the fused features originated
   - developing a new and effective term weighting scheme to assign a generalised weight to topical terms in the collection.
2. The SIF2 model adopts the hybrid fusion strategy to rank local document-specific features and select those that are informative based on their global representativeness. These proposed tactics are achieved by:
   - generating latent topics from indexed paragraphs in the collection
   - adapting the ERS theory of SIF to model more entities from the collection
   - localising the weighting scheme of topical terms based on their appearance in each document and distributing their global topical assignment based on their frequency in the document
   - developing a new and effective term weighting scheme to consider the previous localising process
   - developing a score fusion function that can assign a globally representative score to topical terms.
3. The UR method also adopts the hybrid fusion strategy to reduce uncertainties in relevant features discovered by different TFS models. The proposed steps are to:
   - Generate latent topics from indexed paragraphs in the collection.
   - Adapt the ERS theory of SIF2 to model the exact collection entities.
   - Develop a new relevance function to estimate paragraph level relevance.
   - Develop a weight scaling function to re-rank relevant features discovered by the existing TFS model.

4. The USIF framework adopts multiple fusion models to select and weight representative intra- and inter-cluster features. The proposed steps are to:

    - Cluster relevant documents based on a similarity measure.
    - Develop a line-fitting equation to estimate the number of clusters in a document collection.
    - Generate latent topics from all paragraphs within a cluster.
    - Adapt ERS to model the required entities.
    - Utilise the SIF model to discover important topics in a cluster and facilitate the selection of intra-cluster features.
    - Develop a new UR method to estimate the relevance of all paragraphs in the collection and then utilise it for measuring the thematic significance of topical terms.
    - Develop a conceptual agglomeration technique to select representative inter-cluster topical terms from the discovered clusters.
    - Re-weigh the selected representative terms using the weighting scheme of SIF and combining this in a linear fashion with the UR method and an informative global statistic.

5. The SSIF framework also adopts multiple fusion models to select and weigh specific features. The proposed tactics are to:

    - Generate latent topics from all paragraphs in the relevant document collection.
    - Integrate the UR method with BM25 to reduce uncertainties in relevant documents before training the SVM.
    - Select specific features from both relevant and irrelevant documents in the collection using the SVM.
    - Utilise the SIF model and combine it in a linear fashion with the adapted UR method and an informative global statistic to weigh all topical terms.
    - Re-weigh the re-ranked features of the UR method and the SVM using the weight calculated by the combined SIF, UR method and global statistic.

7.3 Limitations and Future Work

In this section, the limitations of the research presented in this thesis will be discussed and some recommendations for future research outlined.

7.3.1 Limitations

Despite the superior performance of the proposed models and frameworks in selecting features for relevance discovery, these models and frameworks are not free of limitations.

a) **Identical feature set**: This is a common challenge in most feature weighting schemes in which equal weight is assigned to a subset of features (i.e., terms). Equal weighting implies these features have the same degree of relevance even though they are not semantically the same, which also implies the existence of inherited and more complicated type of uncertainties. Tackling this problem by revising the identical set is critical to increase the overall performance of the intended application.

b) **Other types of features**: Only terms and topical statistical features (i.e., term–topic assignment and paragraph–topic distribution) are considered in the fusion strategies adopted in this research. However, other text features, such as patterns, phrases, concepts or a combination of these appear to be beneficial for relevant feature discovery. Incorporating these features into the proposed work might be useful, especially for the selection process.

c) **Advanced clustering algorithms**: Traditional clustering algorithms use distance-based measures to estimate the similarity between documents to form a cluster. These algorithms are (1) only concerned with the spatial relationship between the vectors that represent documents [Li et al., 2016], (2) sensitive to the method of selecting the initial centroids [Li et al., 2016] and (3) unaware of the internal structure of long documents [Shehata et al., 2010]. Thus, using more advanced clustering techniques might help to discover additional interesting features.

d) **Advanced topic modelling**: The proposed models and frameworks used the popular LDA algorithm to extract latent topics. The LDA model forms the basis of many probabilistic topic modelling techniques designed to improve the quality of generated topics. Adopting these enhanced topic models might be more effective in identifying relevant topics or subtopics.

e) **Adding explicit semantics**: The semantic information used in this research is probabilistic and based on the LDA topics. Such semantic information is implicit and usually difficult to interpret [Saif et al., 2016]. Therefore, using explicit semantics (e.g., those based on advanced NLP techniques, ontologies and dictionaries) might aid in understanding the meaning of discovered features and facilitate the selection process.

f) **Introducing parameters**: The proposed models and frameworks did not use any parameters except those of the LDA, clustering and the top-k features. However, it would be practical to introduce certain parameters to control tasks, such as weight optimisation and noise reduction.

g) **More specific features**: It is difficult to define the specificity of features using only relevant documents with SIF, SIF2, USIF and the UR method due to the absence of an explicit query or negative context (i.e., irrelevant documents). Using negative feedback to identify specific relevant features (as done by the SSIF framework) significantly improves the performance of IF and allows the boundary of feature specificity to be defined to some extent. However, if the feature context in both positive and negative feedback is mutually exclusive, identifying specific features is either impossible or ineffective. Thus, it might be useful to introduce an appropriate clustering algorithm to the SSIF framework to delineate a clear boundary between positive and negative feature contexts, which might aid in the selection of specific features.

7.3.2 Future Work

Addressing the limitations outlined in the previous section is the first intended step for future work. Also, the research presented in this thesis can take several future directions, which are noted in this section.

a) Despite the sophistication of the proposed models and frameworks in this thesis and the way they tackle uncertainties in TFS, they still output identical subsets of features (i.e., terms). These sets are problematic and hinder the performance of discriminative algorithms such as IF because it is difficult to differentiate between the elements (i.e., features) of a set. Knowing these elements are semantically different suggests a more comprehensive solution is needed, as this problem is prevalent with almost all TFS techniques. Revising the weight of these elements by integrating granular computing [Yao, 2001] and rough set theory [Yao, 2009] into our ERS theory is a feasible solution. Both granular computing and rough set theory have demonstrated interesting outcomes [Li, 2003, Li and Zhong, 2003] and [Li et al., 2017c, 2012]. Thus, this approach should be investigated in future work.

b) The fusion strategies adopted in the proposed research mainly deal with statistical features (i.e., topics) and lexical words. These features do not consider the sequence of terms as they originally appear in documents and paragraphs. The order of words is important, as it conveys semantic information and discriminates between selected features. In future work, the proposed models and frameworks will be adapted to consider n-grams [Albathan et al., 2013], sequential patterns [Li et al., 2015] or ontological concepts [Tao et al., 2011] to enhance the selection step of relevant features.

c) The document clustering algorithm used in this research has shown remarkable improvement to existing techniques. However, the limitations outlined in the previous section might affect the performance of the USIF framework. It is worth investigating other advanced clustering techniques, such as the collapsed Gibbs sampling algorithm for the Dirichlet multinomial mixture model (GSDMM) [Yin and Wang, 2014] and the constrained heterogeneous information network clustering model (CHINC) [Wang et al., 2015], which are capable of digging deeper into the internal structures of documents or even paragraphs. Also, developing or adapting our intra- and inter-cluster concept agglomeration to select informative features from the newly formed clusters is an essential step forward.

d) The topical features used in this research have been generated by the LDA, which is currently the most widely used unsupervised topic modelling algorithm. However, there are numerous other topic modelling techniques that might generate better quality topics in a supervised or unsupervised way, including the pachinko allocation model (PAM) and the hierarchical pachinko allocation model (hPAM) [Li and McCallum, 2006, Mimno et al., 2007], the segmented topic model (STM) [Du et al., 2010] and the maximum entropy discrimination latent Dirichlet allocation (MedLDA) [Zhu et al., 2012]. It would be beneficial to use these topics knowing that the proposed ERS theory managed to solve many issues of the base topic model (i.e., LDA). Adapting our models and frameworks to the new topics would be useful.

e) Understanding the meaning behind the discovered features would also be useful, especially in the selection process. However, no explicit semantic knowledge is used in our proposed research, and the adopted semantic information is implicit and probabilistically generated.

Thus, adding an explicit semantic layer to the proposed models and frameworks through personalised ontology, advanced NLP methods or a combination of both could dramatically improve our understanding of the topics discussed in the document collection and enhance the selection and weighting of relevant features.

f) In the proposed TFS research, a conservative approach was taken during the training phase of the models and frameworks. No features were ignored except the stop words, as the training documents were relevant. We assumed that all features in these documents were initially relevant. However, it would be practical to introduce control parameters (e.g., hyperparameters) to facilitate issues like noise reduction (e.g., by specifying a cut-off) and weight optimisation. Incorporating parameters into the proposed models and frameworks is feasible and will be considered in future work.

Our proposed models and frameworks can be extended and integrated with potential theories and techniques to explore different research problems. Although the proposed research has been evaluated mainly in the context of IF, it has the potential to be employed in other applications such as IR, recommendation systems, text classification and opinion mining.

1. The proposed techniques have illustrated the capability of fusion strategies to discover relevant features from the contents of relevant and irrelevant documents that represent user information preferences. These techniques can be adapted to similar content-based analysis systems such as the recommender system. For example, the proposed SIF, SIF2 or even the USIF framework could be utilised to select and weigh interesting items from the content of user profiles, and subsequently used to recommend top-k items.

2. No explicit users information needs (e.g., queries) have been assumed to be given in the research work in this thesis. However, for an IR application, such queries can be integrated with the proposed techniques to guide the search for high-quality features. For example, the proposed USIF framework would benefit from the user query to locate the most relevant cluster. Also, an explicit query can be utilised with the proposed UR method to categorise paragraphs in relevant documents based on their specificity to the query, allowing them to be ranked accordingly. In the presence of short queries, the proposed techniques could be used for a query expansion problem.

3. As previously mentioned, IF is regarded as a form of binary classification. Therefore, it

is feasible to adapt our proposed work for incorporation into the related centroid-based or three-way decision methodologies for text classification problems or similar contexts, like sentiment analysis. For instance, our proposed UR method and the SSIF framework significantly improved the performance of the SVM, which indicates the possibility for further improvements in text classification.

Appendix A

Detailed Results: The Proposed SIF Model

Table A.1: Detailed Results of the SIF Model for the First 50 Collections of the RCV1 Dataset

Collection#	nDCG@4	P@20	BP	MAP	$F_{\beta=1}$	IAP	Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
101	0.3868528	1.0000000	0.8631922	0.9288488	0.6514430	0.9199720	0.5016287	1.0000000	1.0000000	1.0000000	0.9831933	0.9802632	0.9537572	0.9547738	0.9004149	0.8711864	0.8028572	0.6732456
102	0.0000000	0.9000000	0.7798742	0.8059042	0.6195130	0.8231076	0.5031447	1.0000000	0.9000000	0.9210526	0.8600000	0.8600000	0.8627451	0.8347826	0.8000000	0.7529412	0.7200000	0.5426621
103	0.2960819	0.7000000	0.4918033	0.5020288	0.5050939	0.5447907	0.5081967	1.0000000	0.7391304	0.7142857	0.7142857	0.4918033	0.4235294	0.4423077	0.4423077	0.4094488	0.4074074	0.2081911
104	0.0000000	0.8500000	0.6276596	0.6456860	0.5669436	0.6627463	0.5053192	1.0000000	0.8846154	0.8846154	0.7435898	0.7031250	0.7014926	0.6354167	0.5454546	0.4331551	0.4093023	0.3494424
105	0.3903800	0.7500000	0.5400000	0.6219502	0.5604391	0.6483184	0.5100000	1.0000000	1.0000000	1.0000000	0.7727273	0.6176471	0.5400000	0.4615385	0.4454545	0.4454545	0.4454545	0.4032258
106	0.3903800	0.1500000	0.0967742	0.1646489	0.2496558	0.2303192	0.5161290	1.0000000	0.1612903	0.1552795	0.1612903	0.1612903	0.1543210	0.1612903	0.1612903	0.1543210	0.1465969	0.1165414
107	0.4692787	0.2000000	0.3513514	0.2442039	0.3309993	0.2644358	0.5135135	0.5000000	0.5000000	0.3750000	0.4166667	0.4166667	0.1775701	0.1455696	0.1092437	0.0994318	0.1000000	0.0686456
108	0.3903800	0.4500000	0.5333334	0.4073960	0.4619349	0.4557604	0.5333334	1.0000000	0.6250000	0.7500000	0.6250000	0.5454546	0.5714286	0.4285714	0.1279070	0.1818182	0.0921053	0.0660793
109	0.2021073	0.9500000	0.4189189	0.5425472	0.5240415	0.5532305	0.5067568	1.0000000	1.0000000	1.0000000	0.4915254	0.4626866	0.3724138	0.3642384	0.3642384	0.3712575	0.3469388	0.3122363
110	0.2463024	0.7500000	0.5483871	0.5387074	0.5271766	0.5928535	0.5161290	1.0000000	0.8125000	0.7857143	0.8125000	0.8125000	0.7619048	0.5581396	0.5581396	0.2795699	0.0765027	0.0639175
111	0.0000000	0.1000000	0.1333333	0.1331301	0.2130731	0.1655131	0.5333334	1.0000000	0.2857143	0.0746269	0.0729927	0.0729927	0.0746269	0.0729927	0.0454545	0.0447761	0.0414201	0.0350467
112	0.9060254	0.5000000	0.5000000	0.6568739	0.5835797	0.6556748	0.5250000	1.0000000	1.0000000	0.8000000	0.7500000	0.7500000	0.5833333	0.6363636	0.6363636	0.4210526	0.3454545	0.2898551
113	0.4692787	0.3000000	0.3428572	0.2690215	0.3515553	0.2926038	0.5071428	0.4242424	0.4242424	0.4516129	0.3818182	0.2886598	0.2681159	0.2485549	0.2425743	0.1866667	0.1720430	0.1301115
114	0.6713861	0.5500000	0.3709678	0.3942677	0.4439904	0.4332674	0.5080645	1.0000000	0.7500000	0.6400000	0.4444445	0.3294118	0.2897196	0.2774194	0.2767296	0.2631579	0.2500000	0.2450593
115	0.4692787	0.8500000	0.6031746	0.7465919	0.6045640	0.7297820	0.5079367	1.0000000	1.0000000	0.9333333	0.9545454	0.8809524	0.7619048	0.8666667	0.7758621	0.4561403	0.2007042	0.1974922
116	0.0000000	0.8500000	0.6666667	0.7344497	0.5990111	0.7434714	0.5057472	1.0000000	0.9166667	0.8285714	0.8055556	0.7777778	0.7121212	0.6896552	0.6853933	0.6730769	0.6165413	0.4728261
117	1.0000000	0.9000000	0.6250000	0.8293392	0.6358950	0.7971186	0.5156250	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.8695652	0.7419355	0.5777778	0.3918919	0.1871345
118	0.5307213	0.3500000	0.4285714	0.4644918	0.4975672	0.4848810	0.5357143	1.0000000	1.0000000	0.5000000	0.5000000	0.5000000	0.4375000	0.5000000	0.3703704	0.3076923	0.1444445	0.0736842
119	0.7039181	0.2000000	0.2750000	0.2841173	0.3655710	0.3460044	0.5125000	1.0000000	0.3055556	0.3055556	0.2772277	0.2772277	0.2772277	0.2772277	0.2772277	0.2720588	0.2700730	0.2666667
120	0.6713861	0.8000000	0.7531645	0.7701278	0.6086599	0.7798477	0.5031645	1.0000000	0.9411765	0.8723404	0.8750000	0.8666667	0.8360656	0.8360656	0.8102190	0.6165049	0.5181159	0.4061697
121	0.8318725	0.9000000	0.7380952	0.7765812	0.6127139	0.7713742	0.5059524	1.0000000	0.9444444	0.9444444	0.8297873	0.8297873	0.8000000	0.7857143	0.7662337	0.6607143	0.6333333	0.2906574
122	0.4981893	0.8500000	0.7450981	0.7061632	0.5921292	0.6926526	0.5098040	1.0000000	1.0000000	0.9230769	0.8695652	0.8620690	0.8529412	0.8378378	0.7254902	0.1923077	0.1835938	0.1722973
123	0.6713861	0.1500000	0.1764706	0.2478195	0.3376050	0.2990831	0.5294118	1.0000000	1.0000000	0.2173913	0.1956522	0.1956522	0.1875000	0.1358025	0.1250000	0.0792079	0.0788177	0.0748899
124	0.6131472	0.1500000	0.1818182	0.1865306	0.2738890	0.2256692	0.5151515	0.6666667	0.1960784	0.2318841	0.1960784	0.1797753	0.1730769	0.1734104	0.1734104	0.1705882	0.1734104	0.1479821
125	0.0000000	1.0000000	0.5151515	0.6232488	0.5571872	0.6201299	0.5037879	1.0000000	0.9600000	0.9090909	0.7407407	0.6309524	0.5178571	0.4938272	0.4656863	0.4435147	0.3644579	0.2953020
126	0.5307213	0.9000000	0.8953489	0.9283363	0.6523933	0.9278924	0.5029069	1.0000000	0.9722222	0.9540230	0.9491525	0.9479167	0.9456522	0.9448819	0.9166667	0.9047619	0.8932585	0.7782806
127	0.0000000	0.7000000	0.5476190	0.5850744	0.5460494	0.6243455	0.5119048	1.0000000	0.7857143	0.8333333	0.6785714	0.6785714	0.5800000	0.5357143	0.5357143	0.4666667	0.4175824	0.3559322
128	0.7653606	0.4500000	0.3636364	0.3840161	0.4400214	0.4491878	0.5151515	1.0000000	0.5384616	0.7000000	0.5384616	0.5384616	0.3673469	0.3278689	0.2727273	0.2700000	0.2112676	0.1764706
129	0.1951900	0.6000000	0.4210526	0.4228632	0.4618566	0.4541660	0.5087720	1.0000000	1.0000000	0.6333333	0.4736842	0.4210526	0.3670886	0.2578616	0.2578616	0.2255319	0.1877256	0.1716868
130	0.7039181	0.2500000	0.3125000	0.3078890	0.3898425	0.3824883	0.5312500	1.0000000	1.0000000	0.5000000	0.2352941	0.2352941	0.2702703	0.2352941	0.2352941	0.1794872	0.1595745	0.1568628
131	0.0000000	0.8500000	0.6486486	0.7233598	0.5959882	0.7330287	0.5067568	1.0000000	0.8947368	0.8823530	0.8947368	0.8947368	0.8695652	0.7741935	0.5714286	0.4960630	0.4331210	0.3523810
132	0.0000000	0.2500000	0.2727273	0.0856993	0.1472564	0.1207630	0.5227273	0.2400000	0.2400000	0.2777778	0.0843373	0.0698413	0.0685358	0.0698413	0.0698413	0.0685358	0.0698413	0.0698413
133	0.6366824	0.4500000	0.4642857	0.5016328	0.5096158	0.5193362	0.5178571	1.0000000	1.0000000	0.8750000	0.8181818	0.4482759	0.4285714	0.3333333	0.2941177	0.2421053	0.1529412	0.1201717
134	0.3903800	0.6000000	0.3432836	0.3310262	0.4006814	0.3911084	0.5074626	1.0000000	0.7777778	0.3898305	0.3150685	0.3076923	0.2614108	0.2653061	0.2653061	0.2614108	0.2653061	0.1930836
135	0.7653606	0.9000000	0.8427300	0.8729712	0.6370246	0.8886224	0.5014837	1.0000000	0.9615384	0.9615384	0.9146342	0.9146342	0.9100529	0.8714860	0.8454810	0.8454810	0.8252688	0.7247312
136	0.2960819	0.4000000	0.3731343	0.3206993	0.3930219	0.3853492	0.5074626	1.0000000	0.3750000	0.4038461	0.3625000	0.3625000	0.3689320	0.3287671	0.3287671	0.3142857	0.2202166	0.1740260
137	0.7653606	0.2500000	0.4444445	0.3760315	0.4484957	0.4285004	0.5555556	1.0000000	1.0000000	0.4444445	0.3000000	0.2857143	0.3571429	0.2727273	0.2727273	0.2352941	0.2727273	0.2727273
138	0.8048100	0.2000000	0.2045455	0.1996002	0.2871265	0.2563788	0.5113636	1.0000000	0.2250000	0.2250000	0.1746032	0.1746032	0.1783784	0.1746032	0.1746032	0.1745283	0.1622642	0.1565836
139	0.9060254	0.6500000	0.7058824	0.6765018	0.5939862	0.7157504	0.5294118	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.7692308	0.5500000	0.5217391	0.6086956	0.2388060	0.1847826
140	1.0000000	0.6500000	0.4477612	0.4639552	0.4847346	0.4745721	0.5074626	1.0000000	0.8750000	0.6400000	0.4897959	0.4576271	0.4320988	0.4056604	0.3455882	0.2030075	0.1889535	0.1825613
141	0.7653606	0.8000000	0.4878049	0.5906197	0.5451016	0.5977781	0.5060976	1.0000000	1.0000000	0.7142857	0.5517241	0.5500000	0.5116279	0.5000000	0.5000000	0.4382716	0.4228571	0.3867925
142	0.4692787	0.3500000	0.2916667	0.3653896	0.4294790	0.4434576	0.5208333	1.0000000	1.0000000	0.8333333	0.4761905	0.4761905	0.1982759	0.1825397	0.1825397	0.1982759	0.1825397	0.1481482
143	0.2960819	0.1000000	0.0869565	0.1121917	0.1846725	0.1650253	0.5217391	1.0000000	0.0844156	0.0821918	0.0844156	0.0844156	0.0821918	0.0805085	0.0805085	0.0801688	0.0782313	0.0782313
144	0.2960819	0.9000000	0.6545454	0.7393518	0.6029869	0.7425133	0.5090909	1.0000000	0.8947368	0.9000000	0.8947368	0.8333333	0.7250000	0.7000000	0.6964286	0.6081081	0.5048544	0.4104478
145	0.2346394	0.1000000	0.1111111	0.0851688	0.1463062	0.0940965	0.5185185	0.1153846	0.1153846	0.0917431	0.0975610	0.0975610	0.0893471	0.0896552	0.0896552	0.0893471	0.0896552	0.0697674
146	0.0000000	0.6500000	0.5495495	0.6302434	0.5604075	0.6548078	0.5045044	1.0000000	0.8297873	0.8297873	0.8297873	0.6923077	0.5648148	0.5367647	0.5155280	0.5174419	0.4755556	0.4111111
147	1.0000000	0.5500000	0.5294118	0.5209063	0.5177875	0.5473980	0.5147059	0.7142857	0.7142857	0.6071429	0.6000000	0.5806451	0.6071429	0.5531915	0.5531915	0.4915254	0.3404255	0.2595420
148	0.0000000	0.8500000	0.9254386	0.9280095	0.6517116	0.9287103	0.5021929	1.0000000	0.9633027	0.9633027	0.9633027	0.9633027	0.9513889	0.9532164	0.9532164	0.9452736	0.9279279	0.6315789
149	0.2021073	0.1000000	0.1754386	0.1730475	0.2582552	0.2100302	0.5087720	0.2524272	0.2524272	0.2523364	0.2524272	0.2524272	0.2102273	0.1777778	0.1777778	0.1777778	0.1777778	0.1269488
150	1.0000000	0.7000000	0.3518519	0.4694999	0.4885721	0.4750203	0.5092593	1.0000000	1.0000000	0.8666667	0.3958333	0.3085106	0.3043478	0.2903226	0.2867647	0.2848101	0.2552083	0.2327586
Avg.	**0.4566359**	**0.5670000**	**0.4751608**	**0.5003752**	**0.4729521**	**0.5268587**	**0.5137146**	**0.9182601**	**0.7569241**	**0.6615829**	**0.5769837**	**0.5413795**	**0.4941959**	**0.4619849**	**0.4303175**	**0.3738135**	**0.3217071**	**0.2582962**

Appendix B

Detailed Results: The Proposed SIF2 Model

Table B.1: Detailed Results of the SIF2 Model for the First 50 Collections of RCV1 Dataset

Collection#	nDCG@4	P@20	BP	MAP	$F_{\beta=1}$	IAP	Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
101	0.3868528	1.0000000	0.8371335	0.9221295	0.6497825	0.9091740	0.5016287	1.0000000	1.0000000	1.0000000	0.9827586	0.9621212	0.9550562	0.9230769	0.8773235	0.8445122	0.8445122	0.6115538
102	0.0000000	0.9500000	0.7798742	0.8347064	0.6278399	0.8374788	0.5031447	1.0000000	0.9500000	0.9016393	0.9016393	0.8854167	0.8854167	0.8738739	0.8129497	0.7527472	0.7272728	0.5213115
103	0.4692787	0.7000000	0.5573770	0.5613163	0.5334373	0.6071813	0.5081967	0.7777778	0.7777778	0.7777778	0.7419355	0.7179487	0.6400000	0.5588235	0.4464286	0.4464286	0.4104478	0.3836478
104	0.0000000	0.9500000	0.6063830	0.6933501	0.5845868	0.6976306	0.5053192	1.0000000	1.0000000	0.9565218	0.8333333	0.7358491	0.6582279	0.6477273	0.5945946	0.4935897	0.4046512	0.3494424
105	0.3903800	0.7000000	0.5800000	0.6632540	0.5766177	0.6808106	0.5100000	1.0000000	1.0000000	1.0000000	0.7600000	0.7407407	0.6046512	0.5769231	0.4814815	0.4555556	0.4347826	0.4347826
106	0.5585076	0.1500000	0.0967742	0.1762388	0.2627561	0.2550958	0.5161290	1.0000000	0.1896552	0.1896552	0.1896552	0.1896552	0.1896552	0.1896552	0.1896552	0.1838235	0.1638418	0.1308017
107	0.4981893	0.2500000	0.2162162	0.2386212	0.3258331	0.2740144	0.5135135	1.0000000	0.8333333	0.2162162	0.1505376	0.1450382	0.1450382	0.1282051	0.1220657	0.1059603	0.0974212	0.0703422
108	0.3903800	0.4000000	0.5333334	0.3928076	0.4524093	0.4279386	0.5333334	1.0000000	0.6666667	0.6000000	0.5333334	0.5333334	0.5333334	0.2812500	0.1896552	0.1621622	0.1333333	0.0742574
109	0.4692787	1.0000000	0.5000000	0.6728416	0.5781071	0.6898621	0.5067568	1.0000000	1.0000000	1.0000000	1.0000000	0.9705882	0.7551020	0.4000000	0.4000000	0.3896104	0.3582888	0.3148936
110	0.3903800	0.8000000	0.8387097	0.6431335	0.5726742	0.7024058	0.5161290	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.0687961	0.0639175
111	0.0000000	0.1500000	0.1333333	0.1405291	0.2224456	0.1657687	0.5333334	1.0000000	0.1818182	0.1500000	0.0937500	0.0937500	0.0731707	0.0731707	0.0488889	0.0365535	0.0365535	0.0357995
112	0.5307213	0.5500000	0.5500000	0.6647796	0.5866789	0.6864402	0.5250000	1.0000000	1.0000000	1.0000000	1.0000000	0.7500000	0.5714286	0.5714286	0.5185185	0.4210526	0.3958333	0.3225806
113	0.6713861	0.4500000	0.4428572	0.3327282	0.4018253	0.3615957	0.5071428	0.5000000	0.4912281	0.4912281	0.4912281	0.4912281	0.3888889	0.2905406	0.2653061	0.2488889	0.1876833	0.1313321
114	0.7653606	0.6000000	0.4516129	0.4405396	0.4718988	0.4664055	0.5080645	0.8461539	0.8461539	0.6363636	0.5263158	0.4912281	0.3522727	0.3057325	0.3057325	0.2880435	0.2864322	0.2460318
115	0.4692787	0.7000000	0.5714286	0.6045777	0.5520597	0.6167222	0.5079367	1.0000000	1.0000000	1.0000000	0.6410257	0.6136364	0.5714286	0.5324675	0.5232558	0.4636364	0.2384937	0.2000000
116	0.0000000	0.8500000	0.7586207	0.8247380	0.6270028	0.8384056	0.5057472	1.0000000	0.9230769	0.9038461	0.9038461	0.9038461	0.9038461	0.8571429	0.8205128	0.7373737	0.6475410	0.6214286
117	0.7653606	0.8500000	0.5937500	0.6914657	0.5907377	0.6766763	0.5156250	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.4285714	0.2839506	0.2745098	0.2636364	0.1927711
118	0.2960819	0.3500000	0.2857143	0.2891648	0.3755937	0.3224022	0.5357143	0.5000000	0.3846154	0.3846154	0.3846154	0.3846154	0.3846154	0.3846154	0.3846154	0.2105263	0.0717949	0.0717949
119	0.5307213	0.5500000	0.3250000	0.4167015	0.4596624	0.4607579	0.5125000	1.0000000	0.8000000	0.6666667	0.5000000	0.3200000	0.3134328	0.3090909	0.3090909	0.3090909	0.3000000	0.2409639
120	0.6713861	0.7500000	0.7468355	0.7837759	0.6128772	0.8026015	0.5031645	1.0000000	0.9000000	0.9000000	0.9000000	0.9000000	0.9000000	0.8807340	0.8496240	0.6614583	0.5274726	0.4093264
121	0.8318725	0.7500000	0.8333333	0.8060027	0.6216661	0.8124495	0.5059524	1.0000000	1.0000000	0.8636364	0.8636364	0.8636364	0.8636364	0.8636364	0.8481013	0.8395062	0.6141732	0.3169811
122	0.7039181	0.8500000	0.8039216	0.8644753	0.6413732	0.8630371	0.5098040	1.0000000	1.0000000	1.0000000	0.8571429	0.8571429	0.8571429	0.8571429	0.8571429	0.8200000	0.7580645	0.6296296
123	0.6713861	0.3000000	0.2941177	0.2911162	0.3756614	0.3448280	0.5294118	1.0000000	1.0000000	0.3157895	0.3157895	0.2682927	0.2682927	0.2682927	0.1237113	0.0797872	0.0765766	0.0765766
124	0.6131472	0.2000000	0.1212121	0.1952800	0.2832047	0.2569791	0.5151515	1.0000000	0.2222222	0.1913043	0.1913043	0.1913043	0.1913043	0.1913043	0.1726619	0.1640212	0.1640212	0.1473214
125	0.0000000	0.8500000	0.5000000	0.5384228	0.5205299	0.5609151	0.5037879	1.0000000	0.8571429	0.6923077	0.6557377	0.6136364	0.5156250	0.4469274	0.3899614	0.3772242	0.3248731	0.2966292
126	0.4981893	0.9500000	0.8953489	0.9310888	0.6530716	0.9351044	0.5029069	1.0000000	0.9777778	0.9777778	0.9701493	0.9594595	0.9453125	0.9453125	0.9453125	0.9012346	0.8959538	0.7678571
127	0.0000000	0.7000000	0.6190476	0.6301055	0.5648882	0.6526714	0.5119048	1.0000000	0.8571429	0.8333333	0.7368421	0.6538461	0.6388889	0.6190476	0.5535714	0.5230770	0.3818182	0.3818182
128	0.7653606	0.2500000	0.3636364	0.2809189	0.3635753	0.3124057	0.5151515	0.3750000	0.3750000	0.3750000	0.3750000	0.3750000	0.3207547	0.2804878	0.2758621	0.2547170	0.2343750	0.1952663
129	0.2463024	0.6500000	0.4385965	0.4963182	0.5024679	0.5268646	0.5087720	1.0000000	1.0000000	0.7200000	0.7200000	0.5476190	0.4531250	0.3627451	0.2985075	0.2598870	0.2429907	0.1906355
130	0.6713861	0.2000000	0.2500000	0.3677687	0.4346453	0.3824965	0.5312500	1.0000000	1.0000000	0.4000000	0.2448980	0.2448980	0.2448980	0.2448980	0.2448980	0.1973684	0.1973684	0.1882353
131	0.2960819	0.9000000	0.7297297	0.8405771	0.6323127	0.8369895	0.5067568	1.0000000	1.0000000	1.0000000	0.9534884	0.9534884	0.9534884	0.8653846	0.7536232	0.7142857	0.5726496	0.4404762
132	0.0000000	0.3000000	0.2727273	0.2347347	0.3239826	0.2627108	0.5227273	1.0000000	0.5555556	0.5555556	0.1794872	0.0856031	0.0856031	0.0856031	0.0856031	0.0856031	0.0856031	0.0856031
133	0.8048100	0.6000000	0.5000000	0.5841202	0.5489964	0.5827838	0.5178571	1.0000000	1.0000000	1.0000000	0.7500000	0.6000000	0.5333334	0.5151515	0.3962264	0.3205128	0.1645570	0.1308411
134	0.3903800	0.6000000	0.5223880	0.5064644	0.5069631	0.5346825	0.5074626	1.0000000	0.6666667	0.6551724	0.5789474	0.5500000	0.5384616	0.4941177	0.4800000	0.4218750	0.3037383	0.1925287
135	0.7653606	1.0000000	0.8397626	0.8823375	0.6395015	0.8852153	0.5014837	1.0000000	0.9500000	0.9078947	0.9078947	0.9078947	0.8855932	0.8855932	0.8637993	0.8489426	0.8374656	0.7422907
136	0.2960819	0.3000000	0.3731343	0.3479407	0.4128273	0.3674176	0.5074626	0.4000000	0.4000000	0.4000000	0.4000000	0.4000000	0.4000000	0.4000000	0.3986014	0.3986014	0.2699115	0.1744792
137	0.7653606	0.2500000	0.2222222	0.2756401	0.3684653	0.3050943	0.5555556	0.4000000	0.4000000	0.4000000	0.3181818	0.3181818	0.3181818	0.3181818	0.3181818	0.2051282	0.1800000	0.1800000
138	0.8318725	0.2500000	0.1590909	0.2501396	0.3359468	0.3014838	0.5113636	1.0000000	0.3333333	0.2530121	0.2530121	0.2530121	0.2527473	0.2195122	0.2035928	0.1956522	0.1941748	0.1582734
139	0.9060254	0.6500000	0.7647059	0.7764128	0.6295518	0.7567821	0.5294118	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.8333333	0.8125000	0.8125000	0.3043478	0.2962963	0.2656250
140	1.0000000	0.9500000	0.5820895	0.6333755	0.5634707	0.6270447	0.5074626	1.0000000	1.0000000	0.9600000	0.9600000	0.7500000	0.6481481	0.5061728	0.4234234	0.2727273	0.1909091	0.1861111
141	0.7653606	0.8000000	0.5000000	0.6259098	0.5596632	0.6320926	0.5060976	1.0000000	1.0000000	0.8181818	0.6744186	0.5500000	0.5232558	0.5177305	0.5177305	0.5177305	0.4662577	0.3677130
142	0.4692787	0.4000000	0.3333333	0.3998019	0.4523619	0.4399978	0.5208333	1.0000000	1.0000000	1.0000000	0.4705882	0.2035398	0.2035398	0.2035398	0.2035398	0.2035398	0.2035398	0.1481482
143	0.2346394	0.1500000	0.1304348	0.0918632	0.1562205	0.1033880	0.5217391	0.1666667	0.1666667	0.1333333	0.1176471	0.0860215	0.0860215	0.0860215	0.0787037	0.0781250	0.0730897	0.0649718
144	0.2960819	0.9000000	0.7090909	0.7439543	0.6045119	0.7633553	0.5090909	1.0000000	0.9130435	0.9130435	0.9130435	0.8888889	0.8055556	0.7551020	0.7222222	0.6428571	0.4950495	0.3481013
145	0.2346394	0.1000000	0.0740741	0.0772089	0.1344045	0.0937063	0.5185185	0.1333333	0.0928270	0.0928270	0.0928270	0.0928270	0.0928270	0.0928270	0.0928270	0.0928270	0.0812500	0.0735695
146	0.0000000	1.0000000	0.6036036	0.7571582	0.6055338	0.7615704	0.5045044	1.0000000	1.0000000	1.0000000	1.0000000	0.9230769	0.8235294	0.6261683	0.5436242	0.5144509	0.4780702	0.4683544
147	1.0000000	0.6000000	0.6470588	0.5383474	0.5262612	0.5861561	0.5147059	0.6571429	0.6571429	0.6571429	0.6571429	0.6571429	0.6571429	0.6571429	0.6410257	0.4838710	0.4050633	0.3177570
148	0.0000000	1.0000000	0.9254386	0.9501113	0.6570788	0.9285288	0.5021929	1.0000000	1.0000000	0.9800000	0.9520958	0.9520958	0.9520958	0.9520958	0.9470588	0.9327354	0.9327354	0.6129032
149	0.2960819	0.1500000	0.2280702	0.1984578	0.2855359	0.2805758	0.5087720	1.0000000	0.2708333	0.2708333	0.2242991	0.2242991	0.1908397	0.1908397	0.1908397	0.1908397	0.1884058	0.1443038
150	1.0000000	0.9500000	0.5740741	0.6616644	0.5755435	0.6701500	0.5092593	1.0000000	0.9500000	0.9500000	0.9500000	0.7500000	0.7500000	0.5500000	0.4712644	0.4190476	0.3798450	0.2014925
Avg.	**0.4721432**	**0.6050000**	**0.5043039**	**0.5353023**	**0.4909009**	**0.5569369**	**0.5137146**	**0.8919965**	**0.7686686**	**0.6986885**	**0.6332260**	**0.5878731**	**0.5500399**	**0.4974052**	**0.4624303**	**0.4117959**	**0.3457523**	**0.2784295**

Appendix C

Detailed Results: The Proposed UR Method

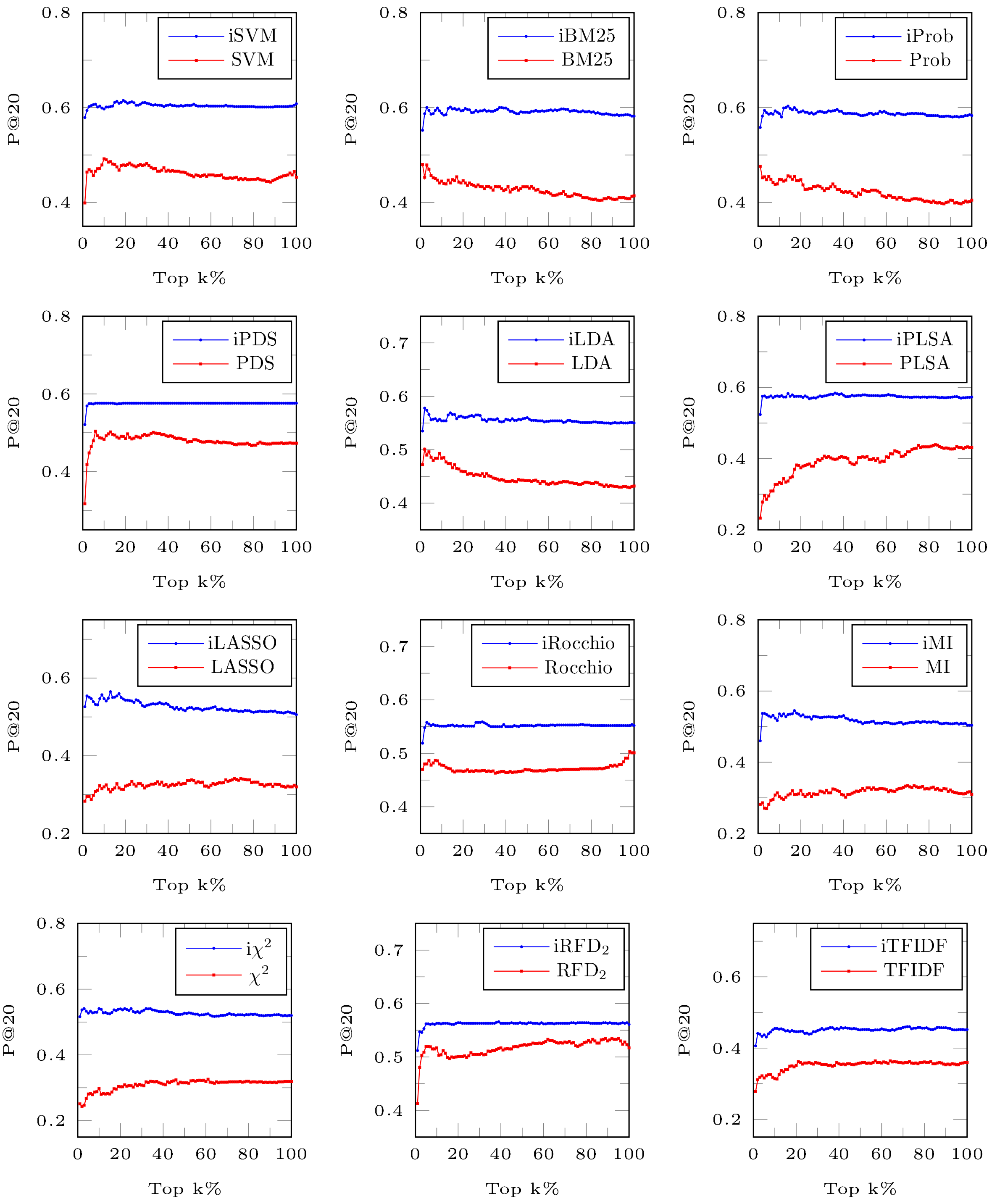


Figure C.1: P@20 Results Before and After Uncertainty Reduction for Each Model from 1% to 100% of the Features Space.

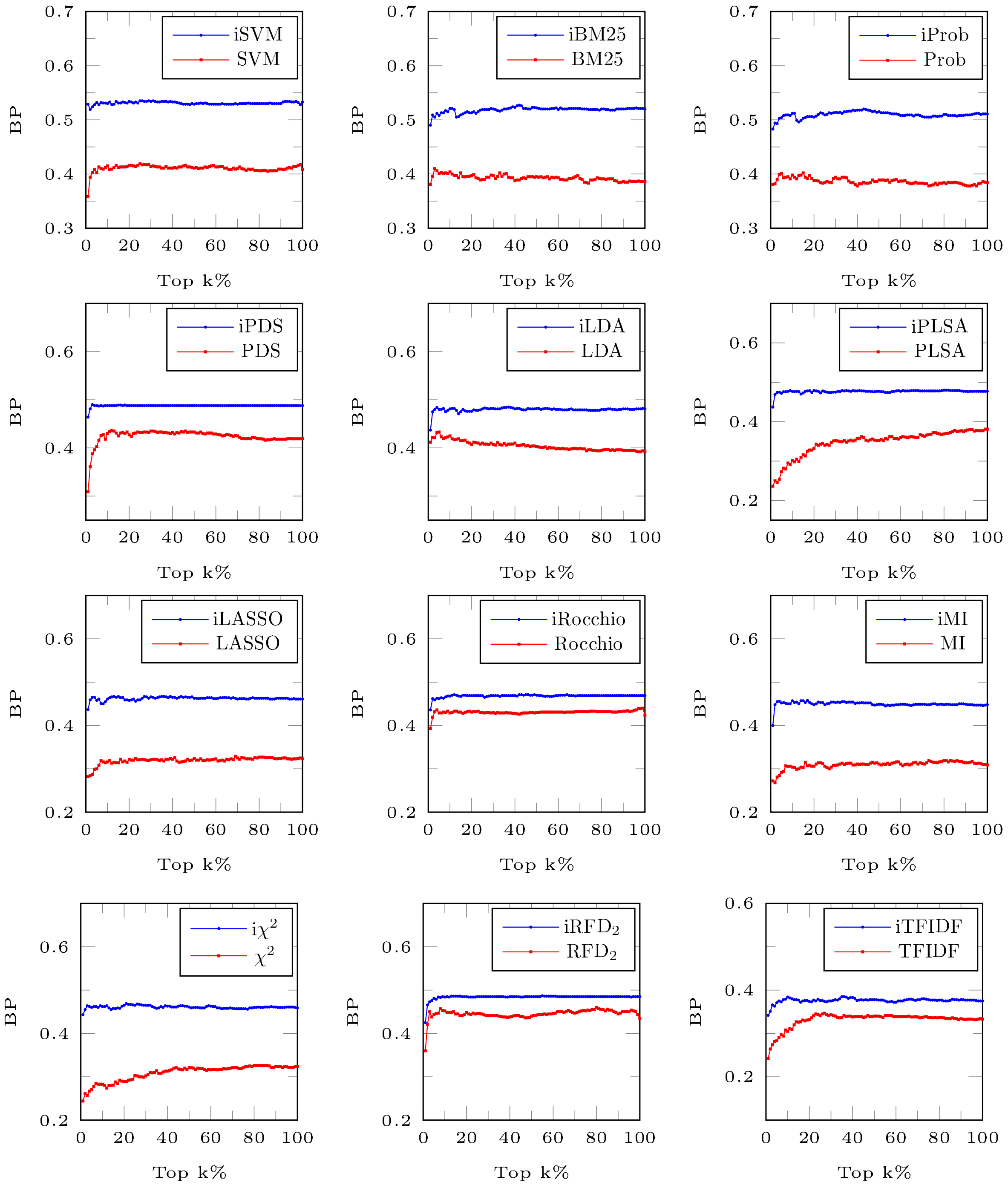


Figure C.2: BP Results Before and After Uncertainty Reduction for Each Model from 1% to 100% of the Features Space.

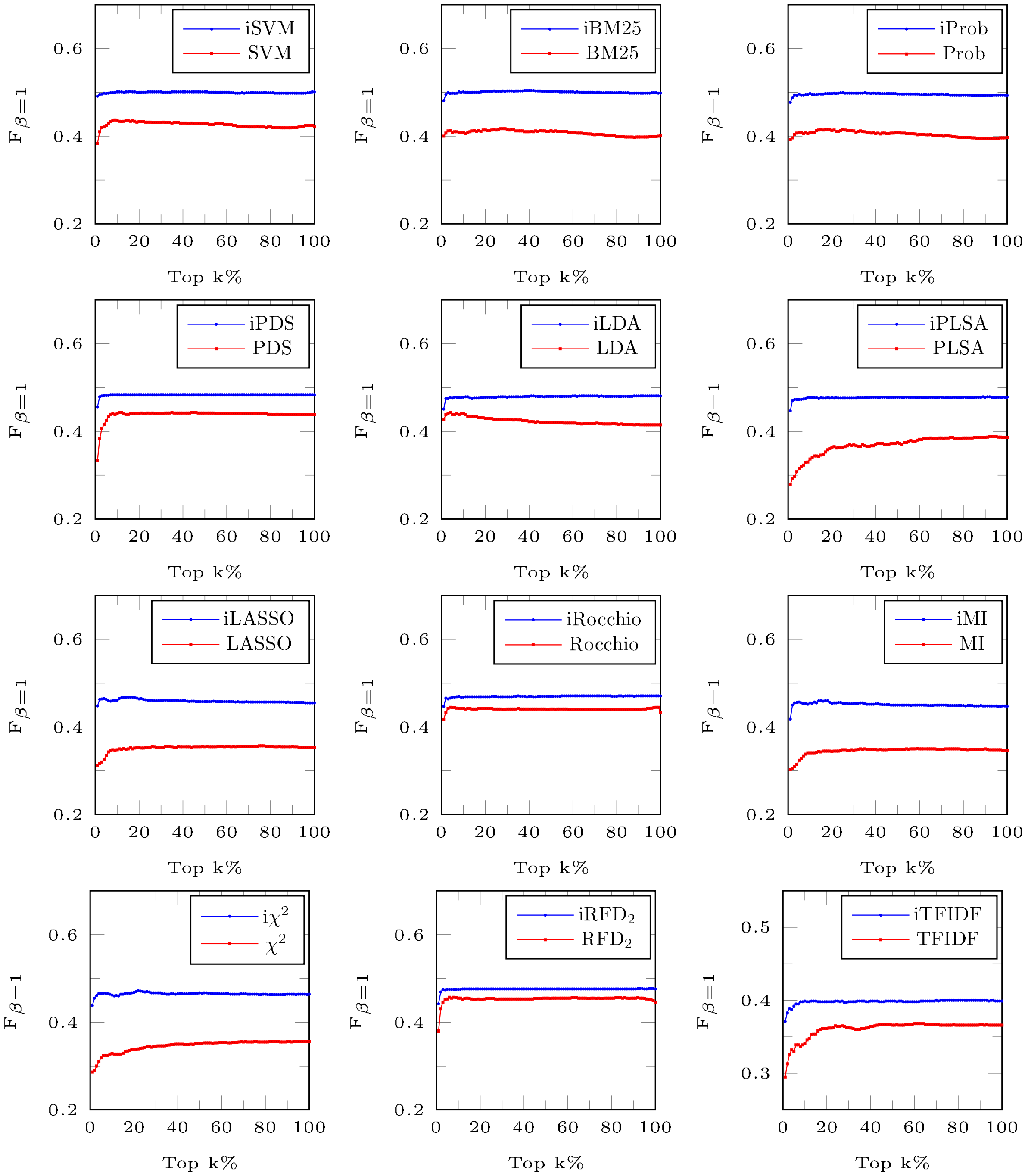


Figure C.3: $F_{\beta=1}$ Results Before and After Uncertainty Reduction for Each Model from 1% to 100% of the Features Space.

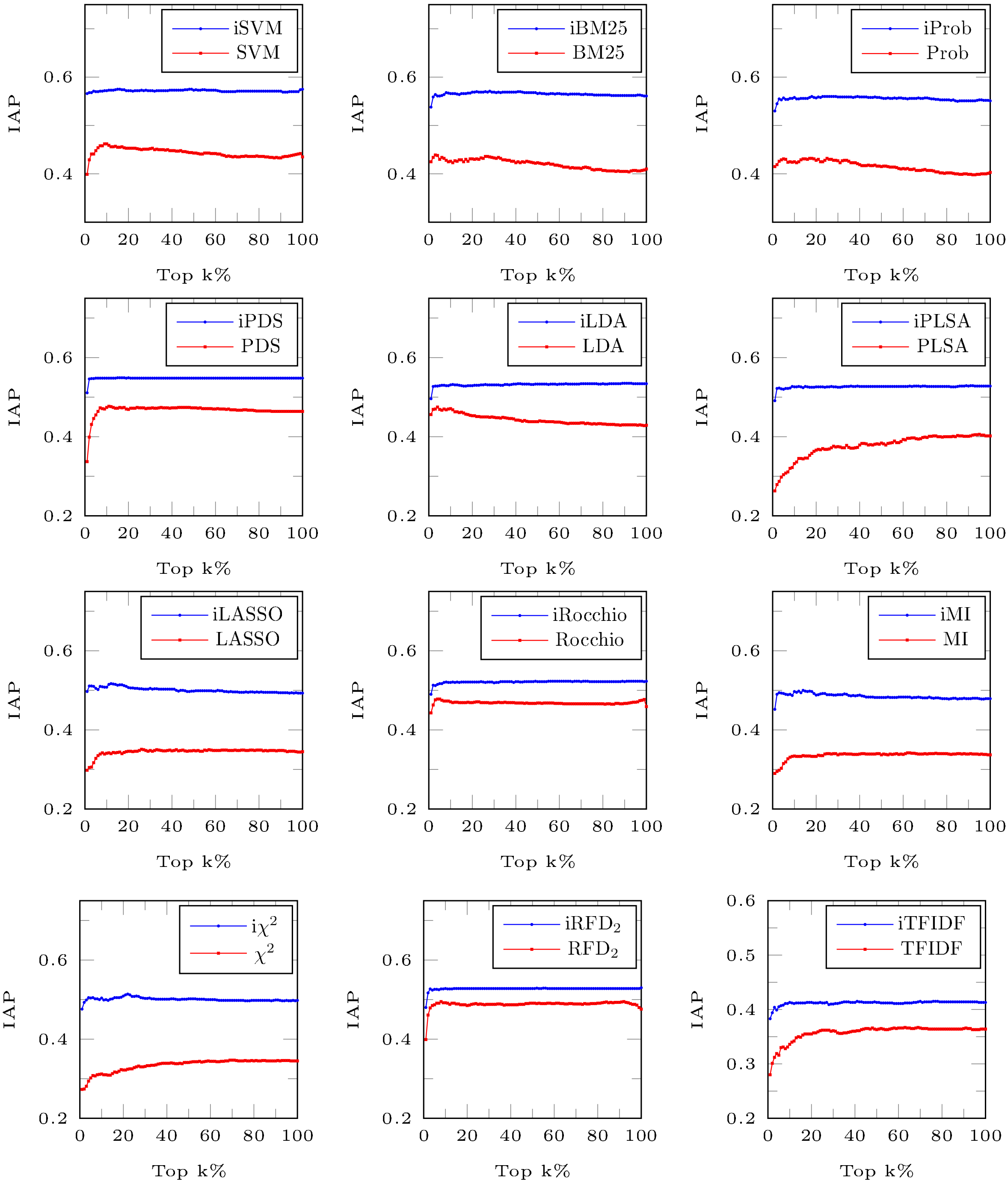


Figure C.4: IAP Results Before and After Uncertainty Reduction for Each Model from 1% to 100% of the Features Space.

Appendix D

Detailed Results: The Proposed USIF Framework

Table D.1: Detailed Results of the USIF Framework for the First 50 Collections of RCV1 Dataset

Collection#	nDCG@4	P@20	BP	MAP	$F_{\beta=1}$	IAP	Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
101	0.6131472	1.0000000	0.8892508	0.9596796	0.6588655	0.9437619	0.5016287	1.0000000	1.0000000	1.0000000	1.0000000	0.9945946	0.9945946	0.9893048	0.9579832	0.9503817	0.8549383	0.6395834
102	0.0000000	0.9000000	0.7798742	0.8428251	0.6301226	0.8353913	0.5031447	1.0000000	0.9444444	0.9200000	0.8985508	0.8846154	0.8709678	0.8608696	0.8115942	0.7513812	0.7272728	0.5196078
103	0.4692787	0.6000000	0.5901640	0.5328653	0.5202387	0.5883213	0.5081967	1.0000000	0.6842105	0.7272728	0.7073171	0.6842105	0.6181818	0.5967742	0.3896104	0.3391813	0.3896104	0.3351648
104	0.0000000	0.9500000	0.6276596	0.7424275	0.6013446	0.7436865	0.5053192	1.0000000	1.0000000	1.0000000	0.9268293	0.9268293	0.7968750	0.6744186	0.6111111	0.4782609	0.4128440	0.3533835
105	0.3903800	0.7000000	0.5800000	0.6856604	0.5849267	0.6983788	0.5100000	1.0000000	1.0000000	1.0000000	0.7500000	0.7500000	0.6279070	0.5882353	0.5362319	0.5000000	0.4752475	0.4545455
106	0.5855701	0.1500000	0.0967742	0.1687288	0.2543180	0.2430861	0.5161290	1.0000000	0.1733333	0.1710526	0.1710526	0.1733333	0.1733333	0.1733333	0.1733333	0.1733333	0.1637427	0.1280992
107	0.6131472	0.3500000	0.2972973	0.3043028	0.3821484	0.3206445	0.5135135	1.0000000	1.0000000	0.2727273	0.2096774	0.2191781	0.2065217	0.1678832	0.1152263	0.1456311	0.1000000	0.0902439
108	0.4692787	0.4000000	0.4666667	0.4909103	0.5112433	0.5263233	0.5333334	1.0000000	1.0000000	0.7500000	0.7500000	0.6363636	0.4705882	0.3666667	0.3666667	0.2500000	0.1238938	0.0753769
109	0.2021073	1.0000000	0.4324324	0.6119888	0.5544236	0.6254494	0.5067568	1.0000000	1.0000000	1.0000000	1.0000000	0.7317073	0.3703704	0.3703704	0.3703704	0.3703704	0.3505155	0.3162393
110	0.3868528	0.8000000	0.7741935	0.6322939	0.5683363	0.6656584	0.5161290	0.8000000	0.8000000	0.8000000	0.8000000	0.8000000	0.8000000	0.8000000	0.7941176	0.7941176	0.0703518	0.0636550
111	0.0000000	0.1000000	0.0666667	0.1431067	0.2256625	0.1704687	0.5333334	1.0000000	0.1363636	0.1333333	0.1333333	0.1363636	0.0849057	0.0849057	0.0472103	0.0391645	0.0397878	0.0397878
112	0.5307213	0.5500000	0.5500000	0.6279055	0.5718602	0.6611460	0.5250000	1.0000000	1.0000000	1.0000000	1.0000000	0.8181818	0.6666667	0.4615385	0.3454545	0.3454545	0.3454545	0.2898551
113	0.4692787	0.3000000	0.4857143	0.3560422	0.4183674	0.3965220	0.5071428	0.5000000	0.5000000	0.5000000	0.5000000	0.5000000	0.4929577	0.4285714	0.3202615	0.2978723	0.1882353	0.1338432
114	0.7653606	0.7000000	0.3709678	0.4219602	0.4610265	0.4582993	0.5080645	0.8888889	0.8888889	0.7777778	0.4750000	0.3250000	0.3027523	0.3014706	0.2795699	0.2795699	0.2753623	0.2470120
115	0.6131472	0.9500000	0.6666667	0.7543512	0.6070923	0.7369163	0.5079367	1.0000000	1.0000000	0.9545454	0.9545454	0.8857143	0.8684211	0.8125000	0.6233766	0.5862069	0.2226563	0.1981132
116	0.0000000	0.8500000	0.7701150	0.8298371	0.6284707	0.8458704	0.5057472	1.0000000	0.9375000	0.9107143	0.9107143	0.9107143	0.9107143	0.8571429	0.8227848	0.7692308	0.6666667	0.6083916
117	0.7653606	0.9500000	0.7187500	0.8162860	0.6320204	0.7966991	0.5156250	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.9545454	0.7419355	0.5777778	0.3333333	0.1560976
118	0.6131472	0.3500000	0.3571429	0.3349426	0.4121796	0.3664669	0.5357143	0.5000000	0.4347826	0.4117647	0.4117647	0.4347826	0.4347826	0.4347826	0.4347826	0.3870968	0.0732984	0.0732984
119	0.7653606	0.5500000	0.4000000	0.4539210	0.4814351	0.4998970	0.5125000	1.0000000	1.0000000	0.6153846	0.5714286	0.4324324	0.3428572	0.3428572	0.3130435	0.3130435	0.3130435	0.2547771
120	0.4692787	0.7500000	0.7594937	0.7576152	0.6047133	0.7953957	0.5031645	0.9042553	0.9042553	0.9000000	0.9010989	0.9042553	0.9042553	0.8909091	0.8538461	0.6464647	0.5306859	0.4093264
121	0.6366824	0.8500000	0.8095238	0.7936722	0.6179636	0.8086052	0.5059524	1.0000000	0.8823530	0.8947368	0.8593750	0.8593750	0.8593750	0.8593750	0.8500000	0.8500000	0.6440678	0.3360000
122	0.7039181	0.8500000	0.7843137	0.7109390	0.5938016	0.6928090	0.5098040	1.0000000	0.9000000	0.8636364	0.8636364	0.8611111	0.8611111	0.8611111	0.8510639	0.1891892	0.1891892	0.1808511
123	0.6713861	0.4000000	0.4705882	0.5716386	0.5497155	0.5606520	0.5294118	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.4074074	0.4074074	0.1363636	0.0727273	0.0727273	0.0705394
124	0.6131472	0.2000000	0.1515152	0.2030395	0.2912766	0.2603640	0.5151515	1.0000000	0.2352941	0.1926606	0.1926606	0.1965812	0.1965812	0.1965812	0.1714286	0.1666667	0.1675676	0.1479821
125	0.0000000	0.9000000	0.5151515	0.5833250	0.5406468	0.6025345	0.5037879	1.0000000	0.9444444	0.8709678	0.6935484	0.6428571	0.5317460	0.4408602	0.4170404	0.4030418	0.3860759	0.2972973
126	0.5307213	0.9500000	0.8953489	0.9267689	0.6520059	0.9260019	0.5029069	1.0000000	0.9777778	0.9772728	0.9666666	0.9615384	0.9304348	0.9304348	0.9000000	0.9000000	0.8908046	0.7510917
127	0.1951900	0.7500000	0.6190476	0.6435965	0.5702463	0.6579581	0.5119048	1.0000000	0.8333333	0.8461539	0.8333333	0.6410257	0.6410257	0.6279070	0.5303030	0.5223880	0.4000000	0.3620690
128	0.7039181	0.3000000	0.3636364	0.3018966	0.3806936	0.3353024	0.5151515	0.4000000	0.4000000	0.4117647	0.4000000	0.4000000	0.3600000	0.3194445	0.3012048	0.2929293	0.2255639	0.1774194
129	0.3903800	0.6000000	0.4561403	0.4742413	0.4909002	0.4846080	0.5087720	1.0000000	1.0000000	0.6190476	0.5882353	0.5106383	0.3717949	0.2923077	0.2697369	0.2658960	0.2418605	0.1711712
130	0.6713861	0.2000000	0.2500000	0.4057498	0.4600953	0.4520497	0.5312500	1.0000000	1.0000000	0.8000000	0.3200000	0.3200000	0.3200000	0.2857143	0.2727273	0.2372881	0.2142857	0.2025317
131	0.2346394	0.9500000	0.6756757	0.8291577	0.6290542	0.8168420	0.5067568	1.0000000	1.0000000	1.0000000	0.9500000	0.9500000	0.9500000	0.9183673	0.6547619	0.6354167	0.5726496	0.3540670
132	0.0000000	0.3000000	0.2727273	0.1835725	0.2717214	0.2409627	0.5227273	1.0000000	0.5555556	0.5000000	0.1076923	0.0696203	0.0696203	0.0696203	0.0696203	0.0696203	0.0696203	0.0696203
133	0.9060254	0.5500000	0.5000000	0.5538344	0.5352419	0.5540635	0.5178571	1.0000000	1.0000000	0.8571429	0.7500000	0.5714286	0.5000000	0.4594595	0.3859649	0.3382353	0.1181818	0.1142857
134	0.4692787	0.6500000	0.4477612	0.4673399	0.4865756	0.5048481	0.5074626	1.0000000	0.6538461	0.6071429	0.6000000	0.5178571	0.4565218	0.4565218	0.4392524	0.3417721	0.2798165	0.2005988
135	1.0000000	0.9000000	0.8308606	0.8426890	0.6287805	0.8641342	0.5014837	1.0000000	0.8793104	0.8906250	0.8906250	0.8647059	0.8529412	0.8529412	0.8529412	0.8358209	0.8333333	0.7522321
136	0.2960819	0.2500000	0.3582090	0.3291611	0.3993121	0.3641232	0.5074626	0.3986014	0.3986014	0.4042553	0.3986014	0.3986014	0.3986014	0.3986014	0.3986014	0.3986014	0.2373541	0.1749347
137	0.7653606	0.2500000	0.4444445	0.4381790	0.4899352	0.5354782	0.5555556	1.0000000	1.0000000	1.0000000	0.7500000	0.5714286	0.3181818	0.3181818	0.3181818	0.2142857	0.2000000	0.2000000
138	0.8318725	0.3500000	0.2272727	0.2797645	0.3616642	0.3151340	0.5113636	1.0000000	0.5555556	0.2500000	0.2315790	0.2346939	0.2346939	0.2196970	0.1968085	0.1968085	0.1877934	0.1588448
139	0.9060254	0.6500000	0.7647059	0.7392536	0.6169784	0.7491651	0.5294118	1.0000000	1.0000000	0.8666667	0.8666667	0.8666667	0.8666667	0.8666667	0.8666667	0.3469388	0.3469388	0.3469388
140	1.0000000	0.9000000	0.5671642	0.6236919	0.5596058	0.6322360	0.5074626	1.0000000	1.0000000	0.9411765	0.9200000	0.8181818	0.6415094	0.5857143	0.4476191	0.2231405	0.1906250	0.1866295
141	0.7653606	0.8500000	0.5000000	0.6200349	0.5573024	0.6346732	0.5060976	1.0000000	1.0000000	0.8500000	0.6500000	0.5873016	0.5384616	0.5376344	0.4963504	0.4963504	0.4807692	0.3445378
142	0.5000000	0.5000000	0.4166667	0.4819365	0.5006306	0.5032898	0.5208333	1.0000000	1.0000000	1.0000000	0.8000000	0.5555556	0.2053572	0.2053572	0.2053572	0.2053572	0.2053572	0.1538462
143	0.2346394	0.1000000	0.0869565	0.0809907	0.1402154	0.0880235	0.5217391	0.1052632	0.1000000	0.0909091	0.0909091	0.0873786	0.0873786	0.0873786	0.0873786	0.0837004	0.0771930	0.0707692
144	0.2346394	0.9000000	0.7454546	0.7757996	0.6147645	0.8003657	0.5090909	1.0000000	0.9230769	0.9230769	0.9230769	0.9230769	0.8297873	0.8297873	0.8297873	0.6521739	0.5882353	0.3819445
145	0.6131472	0.1000000	0.0740741	0.0779815	0.1355737	0.0962691	0.5185185	0.1250000	0.0982659	0.0960699	0.0960699	0.0982659	0.0982659	0.0982659	0.0975610	0.0944206	0.0828026	0.0739726
146	0.0000000	1.0000000	0.5585586	0.7054864	0.5883037	0.7060094	0.5045044	1.0000000	1.0000000	0.9600000	0.9268293	0.6969697	0.6222222	0.5583333	0.5337838	0.5235294	0.4800000	0.4644352
147	0.7653606	0.5500000	0.6764706	0.5377863	0.5259931	0.5753744	0.5147059	0.6857143	0.6857143	0.6944444	0.6944444	0.6857143	0.6857143	0.6857143	0.6857143	0.5800000	0.1322314	0.1137124
148	0.5000000	1.0000000	0.9254386	0.9526129	0.6576760	0.9346949	0.5021929	1.0000000	1.0000000	0.9836066	0.9764706	0.9583333	0.9527027	0.9527027	0.9485714	0.9377990	0.9292035	0.6422535
149	0.3065736	0.2500000	0.2280702	0.1989377	0.2860324	0.2324890	0.5087720	0.2839506	0.2839506	0.2839506	0.2804878	0.2839506	0.2000000	0.2000000	0.2000000	0.2000000	0.2000000	0.1410891
150	0.9197208	0.9000000	0.5925926	0.6762447	0.5809916	0.6851051	0.5092593	1.0000000	0.9375000	0.9333333	0.9000000	0.8888889	0.7297297	0.5789474	0.5066667	0.4423077	0.3983740	0.2204082
Avg.	**0.5017374**	**0.6160000**	**0.5177640**	**0.5495394**	**0.5004499**	**0.5706510**	**0.5137146**	**0.8918335**	**0.7929672**	**0.7290643**	**0.6718444**	**0.6234005**	**0.5531097**	**0.5251623**	**0.4765993**	**0.4142189**	**0.3339912**	**0.2649695**

Appendix E

Detailed Results: The Proposed SSIF Framework

Table E.1: Detailed Results of the SSIF Framework for the First 50 Collections of RCV1 Dataset

Collection#	nDCG@4	P@20	BP	MAP	$F_{\beta=1}$	IAP	Recall	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
101	0.3065736	0.9500000	0.8143323	0.8854953	0.6404472	0.8935572	0.5016287	1.0000000	0.9821429	0.9450000	0.9450000	0.9450000	0.9450000	0.9450000	0.8506944	0.8211920	0.7982954	0.6518047
102	0.4692787	1.0000000	0.8364780	0.9362046	0.6545268	0.9155068	0.5031447	1.0000000	1.0000000	0.9905660	0.9905660	0.9905660	0.9905660	0.9905660	0.9743590	0.8724832	0.7512953	0.5196078
103	0.4692787	0.9500000	0.6393443	0.7596180	0.6089776	0.7576004	0.5081967	1.0000000	1.0000000	1.0000000	1.0000000	0.9615384	0.7804878	0.6491228	0.6231884	0.4833333	0.4833333	0.3526012
104	0.4692787	1.0000000	0.6808510	0.7867288	0.6153783	0.7828135	0.5053192	1.0000000	1.0000000	1.0000000	0.9166667	0.8958333	0.8750000	0.8219178	0.6930693	0.6363636	0.3990826	0.3730159
105	0.0000000	0.7500000	0.5400000	0.6310720	0.5641129	0.6478875	0.5100000	1.0000000	1.0000000	0.9230769	0.8076923	0.8076923	0.5952381	0.5344828	0.5147059	0.3703704	0.3103448	0.2631579
106	0.4414924	0.2000000	0.2258064	0.1609772	0.2454121	0.1861531	0.5161290	0.2333333	0.2333333	0.2333333	0.2187500	0.2187500	0.1550802	0.1550802	0.1550802	0.1550802	0.1550802	0.1347826
107	0.8772153	0.5000000	0.3243243	0.3032707	0.3813335	0.3432863	0.5135135	1.0000000	0.7142857	0.6250000	0.4615385	0.2238806	0.1810345	0.1533333	0.1232227	0.1016949	0.1005747	0.0915842
108	0.0000000	0.3500000	0.4000000	0.3269585	0.4053924	0.3607519	0.5333334	1.0000000	0.6666667	0.6000000	0.4000000	0.4000000	0.2162162	0.1718750	0.1718750	0.1643836	0.1147541	0.0625000
109	0.4692787	1.0000000	0.7432432	0.8155494	0.6250976	0.7994165	0.5067568	1.0000000	1.0000000	1.0000000	1.0000000	0.9705882	0.9069768	0.8823530	0.8666667	0.4460432	0.4060606	0.3148936
110	0.3868528	0.8000000	0.8387097	0.6572248	0.5781935	0.7060801	0.5161290	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.8437500	0.1089494	0.0641822
111	0.0000000	0.2500000	0.2666667	0.2558650	0.3458227	0.3014993	0.5333334	1.0000000	1.0000000	0.3125000	0.3125000	0.2727273	0.1904762	0.0757576	0.0421456	0.0396341	0.0356234	0.0351288
112	1.0000000	0.4500000	0.4500000	0.4163052	0.4643770	0.4627878	0.5250000	1.0000000	1.0000000	0.7777778	0.7777778	0.5294118	0.3750000	0.3750000	0.0773196	0.0614887	0.0614887	0.0554017
113	0.7039181	0.4500000	0.4142857	0.3363194	0.4044330	0.3744662	0.5071428	0.5357143	0.5357143	0.5357143	0.4912281	0.4912281	0.3846154	0.3467742	0.2318182	0.2298387	0.1981424	0.1383399
114	0.7653606	0.5500000	0.4516129	0.4820915	0.4947374	0.4967312	0.5080645	0.8333333	0.7333334	0.5925926	0.5588235	0.5208333	0.4782609	0.4148936	0.4017094	0.3472222	0.3111111	0.2719298
115	0.0000000	0.7000000	0.3492064	0.4903236	0.4989748	0.5359722	0.5079367	1.0000000	1.0000000	1.0000000	0.4313726	0.4166667	0.4166667	0.4166667	0.4166667	0.2830189	0.2830189	0.2316176
116	0.3065736	0.8000000	0.7126437	0.7176548	0.5933485	0.7427136	0.5057472	1.0000000	0.8333333	0.8181818	0.7714286	0.7580645	0.7580645	0.7307692	0.7209302	0.6862745	0.5984849	0.4943182
117	0.7039181	0.9500000	0.7500000	0.8014183	0.6275137	0.7887564	0.5156250	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.9500000	0.9166667	0.8846154	0.4333333	0.2564103	0.2352941
118	0.3065736	0.4000000	0.5000000	0.4117887	0.4656472	0.4318944	0.5357143	1.0000000	1.0000000	0.6666667	0.5333334	0.5333334	0.5333334	0.1149425	0.1149425	0.1142857	0.0700000	0.0700000
119	0.0000000	0.6500000	0.4500000	0.5040684	0.5082492	0.5202687	0.5125000	1.0000000	1.0000000	0.8888889	0.7500000	0.5151515	0.3278689	0.3181818	0.3181818	0.2370370	0.2183908	0.1492537
120	0.7039181	0.8500000	0.7405064	0.8076689	0.6200488	0.8091738	0.5031645	1.0000000	0.8947368	0.8765432	0.8765432	0.8765432	0.8585858	0.8571429	0.7555556	0.7555556	0.7486911	0.4010152
121	0.3903800	0.8000000	0.7023810	0.7460529	0.6029803	0.7641363	0.5059524	1.0000000	0.8333333	0.8307692	0.8307692	0.8307692	0.8307692	0.8307692	0.7763158	0.6915888	0.6525424	0.2978723
122	0.7039181	0.8000000	0.8627451	0.8607025	0.6403319	0.8603120	0.5098040	1.0000000	1.0000000	0.9285714	0.8653846	0.8653846	0.8653846	0.8653846	0.8653846	0.8653846	0.8000000	0.5425532
123	0.0000000	0.3500000	0.4117647	0.3511035	0.4222035	0.3904018	0.5294118	0.6666667	0.6666667	0.6363636	0.6363636	0.6363636	0.3913043	0.2115385	0.1645570	0.1102362	0.0871795	0.0871795
124	0.6131472	0.2500000	0.2121212	0.2506422	0.3372154	0.2713197	0.5151515	0.6666667	0.3333333	0.2758621	0.2758621	0.2758621	0.2531646	0.2531646	0.1678322	0.1677019	0.1657459	0.1493213
125	0.0000000	0.7000000	0.4090909	0.5021803	0.5029828	0.5259179	0.5037879	1.0000000	0.7941176	0.7941176	0.5263158	0.4416667	0.3948340	0.3948340	0.3948340	0.3948340	0.3812500	0.2682927
126	0.7653606	0.9500000	0.9069768	0.9355398	0.6541632	0.9430761	0.5029069	1.0000000	0.9726027	0.9726027	0.9726027	0.9726027	0.9569892	0.9459459	0.9416059	0.9276316	0.9075145	0.8037383
127	0.3903800	0.7500000	0.6190476	0.6117654	0.5573978	0.6166582	0.5119048	1.0000000	0.8888889	0.7619048	0.7619048	0.6279070	0.6279070	0.6279070	0.5454546	0.3673469	0.3482143	0.2258064
128	0.2960819	0.3000000	0.3030303	0.2632897	0.3484761	0.2818924	0.5151515	0.5000000	0.4166667	0.3125000	0.3125000	0.2830189	0.2656250	0.2222222	0.2222222	0.2222222	0.1851852	0.1586538
129	0.5585076	0.9000000	0.6315789	0.6488366	0.5703308	0.6545457	0.5087720	1.0000000	0.9285714	0.9285714	0.9047619	0.6944444	0.6470588	0.6379311	0.5555556	0.4476191	0.3058824	0.1496063
130	0.2346394	0.3500000	0.4375000	0.4916600	0.5106889	0.5098369	0.5312500	1.0000000	1.0000000	1.0000000	0.5555556	0.5384616	0.3200000	0.2666667	0.2666667	0.2542373	0.2542373	0.1523810
131	0.2346394	0.9500000	0.6891892	0.8354376	0.6308530	0.8437507	0.5067568	1.0000000	1.0000000	0.9714286	0.9714286	0.9714286	0.9500000	0.8490566	0.7093023	0.7093023	0.5929204	0.5563910
132	0.0000000	0.2000000	0.1818182	0.2571957	0.3447602	0.2696442	0.5227273	1.0000000	1.0000000	0.2142857	0.2121212	0.0786026	0.0786026	0.0786026	0.0786026	0.0786026	0.0733333	0.0733333
133	0.7039181	0.5000000	0.5000000	0.6334676	0.5698578	0.6327292	0.5178571	1.0000000	1.0000000	1.0000000	0.9090909	0.5172414	0.5172414	0.4791667	0.4791667	0.4791667	0.4561403	0.1228070
134	0.0000000	0.7000000	0.5970149	0.6244991	0.5599305	0.6293111	0.5074626	1.0000000	0.8461539	0.7500000	0.7500000	0.7297297	0.6727273	0.5714286	0.5632184	0.5321101	0.2837209	0.2233333
135	0.0000000	0.8500000	0.7507418	0.8103175	0.6195467	0.8283562	0.5014837	1.0000000	0.9387755	0.9333333	0.8897638	0.8146341	0.8095238	0.7649254	0.7597911	0.7597911	0.7536232	0.6877551
136	0.4692787	0.4000000	0.3880597	0.3982233	0.4462550	0.4191853	0.5074626	0.6666667	0.4666667	0.4255319	0.4252874	0.4252874	0.4252874	0.4141414	0.3984375	0.3875000	0.3875000	0.1887324
137	0.7653606	0.3500000	0.3333333	0.2901000	0.3811639	0.3175641	0.5555556	0.5000000	0.5000000	0.5000000	0.4285714	0.3500000	0.3500000	0.3500000	0.3500000	0.0548781	0.0548781	0.0548781
138	0.1681275	0.7500000	0.5454546	0.5565589	0.5330049	0.5789647	0.5113636	1.0000000	0.8750000	0.8333333	0.8333333	0.7500000	0.6764706	0.5090909	0.3478261	0.2264151	0.1694915	0.1476510
139	0.7653606	0.7000000	0.7647059	0.8641524	0.6565790	0.8585185	0.5294118	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.8666667	0.8666667	0.7777778	0.5925926	0.3400000
140	0.6131472	0.7000000	0.6268657	0.6872808	0.5838397	0.6964436	0.5074626	1.0000000	1.0000000	0.8292683	0.8292683	0.8292683	0.8292683	0.6666667	0.5222222	0.4782609	0.4275862	0.2490706
141	0.4692787	0.5000000	0.5853658	0.5812025	0.5410561	0.6202215	0.5060976	1.0000000	0.9000000	0.5903614	0.5903614	0.5903614	0.5903614	0.5842696	0.5652174	0.5546219	0.4588235	0.3980583
142	1.0000000	0.6000000	0.5833333	0.6695839	0.5859149	0.6778596	0.5208333	1.0000000	1.0000000	1.0000000	1.0000000	0.8333333	0.6153846	0.6153846	0.6071429	0.4347826	0.1965812	0.1538462
143	0.0000000	0.1000000	0.1304348	0.1031015	0.1721785	0.1139308	0.5217391	0.2500000	0.1666667	0.1282051	0.1153846	0.1136364	0.0903226	0.0903226	0.0746753	0.0746753	0.0746753	0.0746753
144	0.0000000	0.5000000	0.5818182	0.5942110	0.5483674	0.6287107	0.5090909	1.0000000	1.0000000	0.6551724	0.6551724	0.6470588	0.6400000	0.6065574	0.6029412	0.5238096	0.2925532	0.2925532
145	0.6131472	0.0500000	0.1111111	0.0844872	0.1452994	0.1061500	0.5185185	0.2500000	0.1111111	0.0934066	0.0934066	0.0934066	0.0934066	0.0934066	0.0909091	0.0879121	0.0819672	0.0787172
146	0.0000000	1.0000000	0.8558559	0.9388253	0.6563178	0.9297459	0.5045044	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.9750000	0.9750000	0.9673913	0.7575758	0.5522388
147	1.0000000	0.6500000	0.6176471	0.6042943	0.5559138	0.6326666	0.5147059	0.8750000	0.8750000	0.8750000	0.6818182	0.6818182	0.6451613	0.6388889	0.6315789	0.5714286	0.3595506	0.1240876
148	0.6309298	1.0000000	0.9254386	0.9576538	0.6588732	0.9399759	0.5021929	1.0000000	1.0000000	1.0000000	0.9857143	0.9568346	0.9568346	0.9529412	0.9529412	0.9497488	0.9276596	0.6570605
149	0.6131472	0.4000000	0.3859649	0.3742748	0.4312807	0.4119870	0.5087720	1.0000000	0.4411765	0.4411765	0.4318182	0.3870968	0.3734940	0.3448276	0.3448276	0.3430657	0.2886598	0.1357143
150	0.6131472	0.9500000	0.7407407	0.7955762	0.6210048	0.7861273	0.5092593	1.0000000	0.9729730	0.9729730	0.9729730	0.9729730	0.9729730	0.9729730	0.9090909	0.4489796	0.2500000	0.2014925
Avg.	**0.4198282**	**0.6310000**	**0.5503828**	**0.5761764**	**0.5146158**	**0.5919451**	**0.5137146**	**0.8964226**	**0.8273000**	**0.7456866**	**0.6900901**	**0.6416150**	**0.5920463**	**0.5478991**	**0.5102097**	**0.4394281**	**0.3595343**	**0.2611646**

Appendix F

TREC Topics of RCV1 Collections

Collection#	**Topic title**
101	Economic Espionage
102	Convicts, Repeat Offenders
103	Ferry Boat Sinkings
104	Rescue of Kidnapped Children
105	Sport Utility Vehicles U.S.
106	Government Supported School Vouchers
107	Tourism Great Britain
108	Harmful Weight-loss Drugs
109	Child custody cases
110	Terrorism Middle East Tourism
111	Telemarketing Practices U.S.
112	School Bus Accidents
113	Ford Foreign Ventures
114	Effects of Global Warming
115	Indian Casino Laws
116	Archaeology Discoveries
117	Organ Transplants in the UK
118	Progress in Treatment of Schizophrenia
119	U.S. Gas Prices
120	Deaths Mining Accidents
121	China Pakistan Nuclear Missile
122	Symptoms Parkinson's Disease
123	Newspaper Circulation Decline
124	Aborigine Health
125	Scottish Independence
126	Nuclear Plants U.S.
127	U.S. Automobile Seat Belt
128	Child Labor Laws
129	Problems Illegal Aliens U.S.
130	College Tuition Planning
131	Television U.S. Children
132	Friendly Fire Deaths
133	Anti-rejection Transplant Drugs
134	Crime Statistics Great Britain
135	WTO Trade Debates
136	Substance Abuse Crime
137	Sea Turtle Deaths
138	Creutzfeldt-Jakob, Mad Cow Disease
139	Pig Organ Transplants
140	Computer Simulation
141	Environment National Park
142	Illiteracy Arab Africa
143	Improving Aircraft Safety
144	Mountain Climbing Deaths
145	Airline Passenger Disruptions
146	Germ Warfare
147	Natural Gas Vehicles
148	NAFTA
149	Aid to Handicapped People
150	Drive-by Shootings

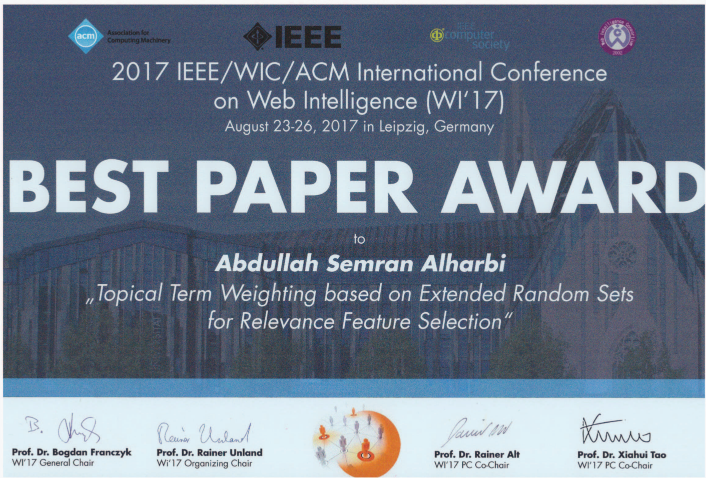
Appendix G

Stop-Words List

a, a’s, able, about, above, according, accordingly, across, actually, after, afterwards, again, against, ain’t, all, allow, allows, almost, alone, along, already, also, although, always, am, among, amongst, an, and, another, any, anybody, anyhow, anyone, anything, anyway, anyways, anywhere, apart, appear, appreciate, appropriate, are, aren’t, around, as, aside, ask, asking, associated, at, available, away, awfully, b, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, believe, below, beside, besides, best, better, between, beyond, both, brief, but, by, c, c’mon, c’s, came, can, can’t, cannot, cant, cause, causes, certain, certainly, changes, clearly, co, com, come, comes, concerning, consequently, consider, considering, contain, containing, contains, corresponding, could, couldn’t, course, currently, d, definitely, described, despite, did, didn’t, different, do, does, doesn’t, doing, don’t, done, down, downwards, during, e, each, edu, eg, eight, either, else, elsewhere, enough, entirely, especially, et, etc, even, ever, every, everybody, everyone, everything, everywhere, ex, exactly, example, except, f, far, few, fifth, first, five, followed, following, follows, for, former, formerly, forth, four, from, further, furthermore, g, get, gets, getting, given, gives, go, goes, going, gone, got, gotten, greetings, h, had, hadn’t, happens, hardly, has, hasn’t, have, haven’t, having, he, he’s, hello, help, hence, her, here, here’s, hereafter, hereby, herein, hereupon, hers, herself, hi, him, himself, his, hither, hopefully, how, howbeit, however, i, i’d, i’ll, i’m, i’ve, ie, if, ignored, immediate, in, inasmuch, inc, indeed, indicate, indicated, indicates, inner, insofar, instead, into, inward, is, isn’t, it, it’d, it’ll, it’s, its, itself, j, just, k, keep, keeps, kept, know, knows, known, l, last, lately, later, latter, latterly, least, less, lest, let, let’s, like, liked, likely, little, look, looking, looks, ltd, m, mainly, many, may, maybe, me, mean, meanwhile, merely, might, more, moreover, most, mostly, much, must, my, myself, n, name, namely, nd, near, nearly, necessary, need, needs, neither, never, nevertheless, new, next, nine, no, nobody, non, none, noone, nor, normally, not, nothing, novel, now, nowhere, o, obviously, of, off, often, oh, ok, okay, old, on, once, one, ones, only, onto, or, other, others, otherwise, ought, our, ours, ourselves, out, outside, over, overall, own, p, particular, particularly, per, perhaps, placed, please, plus, possible, presumably, probably, provides, q, que, quite, qv, r, rather, rd, re, really, reasonably, regarding, regardless, regards, relatively, respectively, right, s, said, same, saw, say, saying, says, second, secondly, see, seeing, seem, seemed, seeming, seems, seen, self, selves, sensible, sent, serious, seriously, seven, several, shall, she, should, shouldn’t, since, six, so, some, somebody, somehow, someone, something, sometime, sometimes, somewhat, somewhere, soon, sorry, specified, specify, specifying, still, sub, such, sup, sure, t, t’s, take, taken, tell, tends, th, than, thank, thanks, thanx, that, that’s, thats, the, their, theirs, them, themselves, then, thence, there, there’s, thereafter, thereby, therefore, therein, theres, thereupon, these, they, they’d, they’ll, they’re, they’ve, think, third, this, thorough, thoroughly, those, though, three, through, throughout, thru, thus, to, together, too, took, toward, towards, tried, tries, truly, try, trying, twice, two, u, un, under, unfortunately, unless, unlikely, until, unto, up, upon, us, use, used, useful, uses, using, usually, uucp, v, value, various, very, via, viz, vs, w, want, wants, was, wasn’t, way, we, we’d, we’ll, we’re, we’ve, welcome, well, went, were, weren’t, what, what’s, whatever, when, whence, whenever, where, where’s, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, who’s, whoever, whole, whom, whose, why, will, willing, wish, with, within, without, won’t, wonder, would, wouldn’t, x, y, yes, yet, you, you’d, you’ll, you’re, you’ve, your, yours, yourself, yourselves, z, zero

Appendix H

Research Awards



2017 IEEE/WIC/ACM International Conference on Web Intelligence (WI'17)

August 23-26, 2017 in Leipzig, Germany

BEST PAPER AWARD

to

Abdullah Semran Alharbi

„Topical Term Weighting based on Extended Random Sets for Relevance Feature Selection"

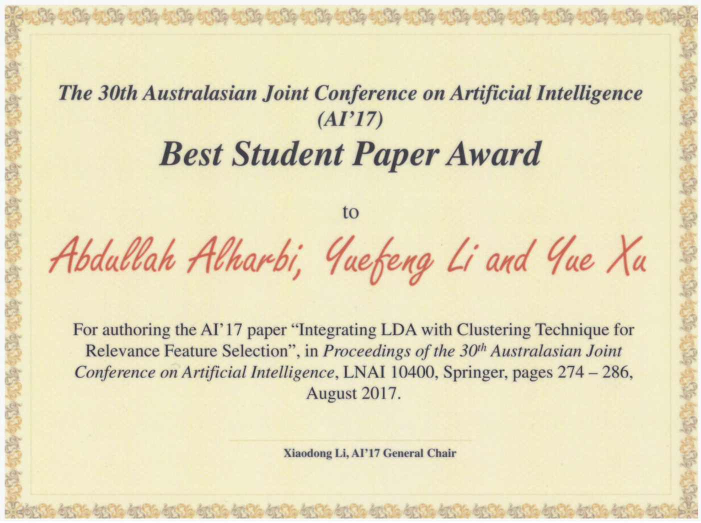
Prof. Dr. Bogdan Franczyk
WI'17 General Chair

Prof. Dr. Rainer Unland
WI'17 Organizing Chair

Prof. Dr. Rainer Alt
WI'17 PC Co-Chair

Prof. Dr. Xiahui Tao
WI'17 PC Co-Chair

Figure H.1: WI 2017 Best Paper Award



The 30th Australasian Joint Conference on Artificial Intelligence (AI'17)

Best Student Paper Award

to

Abdullah Alharbi, Yuefeng Li and Yue Xu

For authoring the AI'17 paper "Integrating LDA with Clustering Technique for Relevance Feature Selection", in *Proceedings of the 30th Australasian Joint Conference on Artificial Intelligence*, LNAI 10400, Springer, pages 274 – 286, August 2017.

Xiaodong Li, AI'17 General Chair

Figure H.2: AI 2017 Best Student Paper Award

Literature Cited

Abul Bashar, M. (2017). *A Personalised Ontology Framework for Interpreting Discovered Knowledge in Text Information*. Phd thesis, Queensland University of Technology.

Aggarwal, C. C. and Zhai, C. (2012). A Survey of Text Clustering Algorithms. In Charu C. Aggarwal and Zhai, C., editors, *Mining Text Data*, pages 77–128. Springer Science+Business Media.

Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In *Proceedings of 20th International Conference on Very Large Data Bases (VLDB'94)*, pages 487–499, Santiago de Chile, Chile. Morgan Kaufmann.

Albathan, M., Li, Y., and Algarni, A. (2012). Using Patterns Co-occurrence Matrix for Cleaning Closed Sequential Patterns for Text Mining. In *Proceedings of the 2012 IEEE/WIC/ACM International Conferences on Web Intelligence (WI'12)*, pages 201–205, Macau, China. IEEE Computer Society.

Albathan, M., Li, Y., and Algarni, A. (2013). Enhanced N-Gram Extraction Using Relevance Feature Discovery. In Cranefield, S. and Nayak, A., editors, *AI 2013: Advances in Artificial Intelligence*, pages 453–465. Springer, Cham.

Albathan, M., Li, Y., and Xu, Y. (2014). Using Extended Random Set to Find Specific Patterns. In *Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, pages 30–37, Warsaw, Poland. IEEE.

Albathan, M. M. M. (2015). *Enhancement of relevant features for text mining*. PhD thesis, Queensland University of Technology, Brisbane, Australia.

Algarni, A. (2011). *Relevance Feature Discovery for Text Analysis*. Phd thesis, Queensland University of Technology.

Algarni, A. (2014). Mining Positive Relevance Feedback to Determine User Information Needs. *Arabian Journal for Science and Engineering*, 39(12):8765–8774.

Algarni, A. and Li, Y. (2013). Mining Specific Features for Acquiring User Information Needs. In Pei, J., Tseng, V. S., Cao, L., Motoda, H., and Xu, G., editors, *Advances in Knowledge Discovery and Data Mining. PAKDD 2013*, pages 532–543, Gold Coast, Australia. Springer Berlin Heidelberg.

Algarni, A., Li, Y., and Xu, Y. (2010). Selected New Training Documents to Update User Profile. In *Proceedings of the 19th ACM international conference on Information and knowledge management (CIKM'10)*, pages 799–808, Toronto, ON, Canada.

Alharbi, A. S., Bashar, A., and Li, Y. (2018a). Random-Sets for Dealing with Uncertainties in Relevance Feature. In Mitrovic, T., Xue, B., and Li, X., editors, *AI 2018: Advances in Artificial Intelligence, Proceedings of the 31st Australasian Joint Conference on Artificial Intelligence*, pages 656–668, Wellington, New Zealand. Springer.

Alharbi, A. S., Li, Y., and Xu, Y. (2017a). Enhancing Topical Word Semantic for Relevance Feature Selection. In Kanagasabai, R., Morshed, A., and Purohit, H., editors, *Proceedings of IJCAI Workshop on Semantic Machine Learning (SML 2017)*, pages 27–33, Melbourne, Australia. CEUR.

Alharbi, A. S., Li, Y., and Xu, Y. (2017b). Integrating LDA with Clustering Technique for Relevance Feature Selection. In Peng, W., Alahakoon, D., and Li, X., editors, *AI 2017: Advances in Artificial Intelligence, Proceedings of the 30th Australasian Joint Conference on Artificial Intelligence*, pages 274–286, Melbourne, VIC, Australia. Springer.

Alharbi, A. S., Li, Y., and Xu, Y. (2017c). Topical Term Weighting based on Extended Random Sets for Relevance Feature Selection. In *Proceedings of the International Conference on Web Intelligence (WI'17)*, pages 654–661, Leipzig, Germany. ACM Press.

Alharbi, A. S., Li, Y., and Xu, Y. (2018b). An Extended Random-Sets Model for Fusion-Based Text Feature Selection. In Phung, D., Tseng, V. S., Webb, G. I., Ho, B., Ganji, M., and Rashidi, L., editors, *Advances in Knowledge Discovery and Data Mining Part III*, *Proceedings of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2018)*, pages 126–138, Melbourne, Australia. Springer, Cham.

Alqhtani, S. M., Luo, S., and Regan, B. (2018). A multiple kernel learning based fusion for earthquake detection from multimedia twitter data. *Multimedia Tools and Applications*, 77(10):12519–12532.

Anastasiu, D., Tagarelli, A., and Karypis, G. (2013). Document Clustering: The Next Frontier. In Aggarwal, C. C. and Reddy, C. K., editors, *Data Clustering: Algorithms and Applications*, pages 305–338. CRC Press Taylor & Francis Group.

Anava, Y., Shtok, A., Kurland, O., and Rabinovich, E. (2016). A Probabilistic Fusion Framework. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management (CIKM '16)*, pages 1463–1472, Indianapolis, IN, USA. ACM Press.

Aphinyanaphongs, Y., Fu, L. D., Li, Z., Peskin, E. R., Efstathiadis, E., Aliferis, C. F., and Statnikov, A. (2014). A Comprehensive Empirical Comparison of Modern Supervised Classification and Feature Selection Methods for Text Categorization. *Journal of the Association for Information Science and Technology*, 65(10):1964–1987.

Atrey, P. K., Hossain, M. A., Saddik, A. E., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis : a survey. *Multimedia Systems*, 16(6):345–379.

Balazs, J. A. and Velásquez, J. D. (2016). Opinion Mining and Information Fusion: A survey. *Information Fusion*, 27:95–110.

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. (2019). Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443.

Bashar, M. A. and Li, Y. (2017). Random Set to Interpret Topic Models in Terms of Ontology Concepts. In Peng, W., Alahakoon, D., and Li, X., editors, *AI 2017: Advances in Artificial Intelligence*, pages 237–249, Melbourne, Australia. Springer, Cham.

Bashar, M. A. and Li, Y. (2018). Interpretation of text patterns. *Data Mining and Knowledge Discovery*, 32(4):849–884.

Bashar, M. A., Li, Y., and Gao, Y. (2016). A Framework for Automatic Personalised Ontology Learning. In *Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2016)*, pages 105–112, Omaha, NE, USA. IEEE Computer Society.

Bashar, M. A., Li, Y., Shen, Y., and Albathan, M. (2014). Interpreting Discovered Patterns in Terms of Ontology Concepts. In *2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, volume 1, pages 432–437. IEEE.

Bashar, M. A., Li, Y., Shen, Y., Gao, Y., and Huang, W. (2017). Conceptual annotation of text patterns. *Computational Intelligence*, 33(4):948–979.

Beil, F., Ester, M., and Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pages 436–442, Edmonton, Alberta, Canada. ACM, ACM.

Belkin, N. J. and Croft, W. B. (1992). Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Communications of the ACM (CACM)*, 35(12):29–38.

Bendersky, M. and Croft, W. B. (2008). Discovering Key Concepts in Verbose Queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'08)*, page 491, Singapore, Singapore. ACM.

Bendersky, M. and Croft, W. B. (2012). Modeling Higher-Order Term Dependencies in Information Retrieval using Query Hypergraphs. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'12)*, page 941, Portland, OR, USA. ACM.

Bendersky, M. and Kurland, O. (2010). Utilizing passage-based language models for ad hoc document retrieval. *Information Retrieval*, 13(2):157–187.

Bendersky, M., Metzler, D., and Croft, W. B. (2011). Parameterized Concept Weighting in Verbose Queries. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*, pages 605–614, Beijing, China.

Bharath Bhushan, S. and Danti, A. (2017). Classification of text documents based on score level fusion approach. *Pattern Recognition Letters*, 94:118–126.

Bing, L., Jiang, S., Lam, W., Zhang, Y., and Jameel, S. (2015). Adaptive Concept Resolution for document representation and its applications in text mining. *Knowledge-Based Systems*, 74:1–13.

Blei, D., Carin, L., and Dunson, D. (2010a). Probabilistic Topic Models: A focus on graphical model design and applications to document and image analysis. *IEEE Signal Processing Magazine*, 27(6):55–65.

Blei, D. M. (2012). Probabilistic Topic Models. *Communications of the ACM*, 55(4):77–84.

Blei, D. M., Griffiths, T. L., and Jordan, M. I. (2010b). The Nested Chinese Restaurant Process and Bayesian Nonparametric Inference of Topic Hierarchies. *Journal of the ACM (JACM)*, 57(2):7:1–7:30.

Blei, D. M. and Lafferty, J. D. (2009). Topic models. *Text mining: classification, clustering, and applications*, 10(71):34.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet Allocation. *The Journal of Machine Learning Research*, 3:993–1022.

Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519.

Buckley, C. and Voorhees, E. M. (2000). Evaluating Evaluation Measure Stability. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000)*, pages 33–40, Athens, Greece. ACM.

Cai, D., Zhang, C., and He, X. (2010). Unsupervised Feature Selection for Multi-Cluster Data. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, pages 333–342, Washington, DC, USA. ACM.

Cai, L. and Hofmann, T. (2004). Hierarchical Document Categorization with Support Vector Machines. In *Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management (CIKM'04)*, pages 78–87, Washington, DC, USA. ACM.

Callan, J. P. (1994). Passage-Level Evidence in Document Retrieval. In Croft, W. B. and van Rijsbergen, C. J., editors, *Proceedings of the 17th Annual International ACM-SIGIR*

Conference on Research and Development in Information Retrieval (SIGIR'94), pages 302–310, Dublin, Ireland. ACM/Springer.

Chaney, A. J. B. and Blei, D. M. (2012). Visualizing Topic Models. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, pages 419–422, Dublin, Ireland. The AAAI Press.

Chemudugunta, C., Holloway, A., Smyth, P., and Steyvers, M. (2008). Modeling Documents by Combining Semantic Concepts with Unsupervised Statistical Learning. In Sheth, A. P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T. W., and Thirunarayan, K., editors, *Proceedings of the 7th International Semantic Web Conference (ISWC 2008)*, pages 229–244, Karlsruhe, Germany. Springer.

Chen, K., Zhang, Z., Long, J., and Zhang, H. (2016). Turning from TF-IDF to TF-IGM for term weighting in text classification. *Expert Systems with Applications*, 66:245–260.

Chen, M., Wang, S., Liang, P. P., Baltrušaitis, T., Zadeh, A., and Morency, L.-P. (2017). Multimodal Sentiment Analysis with Word-Level Fusion and Reinforcement Learning. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction (ICMI 2017)*, pages 163–171, New York, New York, USA. ACM Press.

Chen, Y.-T. and Chen, M. C. (2011). Using chi-square statistics to measure similarities for text categorization. *Expert Systems with Applications*, 38(4):3085–3090.

Chien, J.-T. (2016). Hierarchical Theme and Topic Modeling. *IEEE Transactions on Neural Networks and Learning Systems*, 27(3):565–578.

Chuang, J., Gupta, S., Manning, C. D., and Heer, J. (2013). Topic Model Diagnostics: Assessing Domain Relevance via Topical Alignment. In *Proceedings of the 30th International Conference on Machine Learning (ICML 2013)*, pages 612–620, Atlanta, GA, USA.

Combarro, E. F., Montanes, E., Diaz, I., Ranilla, J., and Mones, R. (2005). Introducing a Family of Linear Measures for Feature Selection in Text Categorization. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1223–1232.

Creswell, J. W. (2013). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications, 4 edition.

Croft, W. B. (2000). Combining Approaches to Information Retrieval. In Croft, W. B., editor, *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*, pages 1–36. Springer US, Boston, Massachusetts, USA.

Cummins, R. and O'riordan, C. (2005). Evolving General Term-Weighting Schemes for Information Retrieval: Tests on Larger Collections. *Artificial Intelligence Review*, 24(3-4):277–299.

Cummins, R. and O'Riordan, C. (2006). Evolving local and global weighting schemes in information retrieval. *Information Retrieval*, 9(3):311–330.

Dang, E. K. F., Luk, R. W., and Allan, J. (2015). A Context-Dependent Relevance Model. *Journal of the Association for Information Science and Technology*, 3(2):80–90.

Das, S., Abraham, A., and Konar, A. (2008). Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, 38(1):218–237.

Dasgupta, A., Drineas, P., Harb, B., Josifovski, V., and Mahoney, M. W. (2007). Feature Selection Methods for Text Classification. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 230–239. ACM.

Datta, D., Singh, S. K., and Chowdary, C. R. (2017). Bridging the gap: effect of text query reformulation in multimodal retrieval. *Multimedia Tools and Applications*, 76(21):22871–22888.

Datta, D., Varma, S., Ravindranath Chowdary, C., and Singh, S. K. (2016). Multimodal Retrieval using Mutual Information based Textual Query Reformulation. *Expert Systems with Applications*, 68:81–92.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., and Harshman, R. A. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Dhar, V. (2013). Data Science and Prediction. *Communications of the ACM*, 56(12):64–73.

Ding, Y. and Yan, S. (2015). Topic Optimization Method Based on Pointwise Mutual Information. In Arik, S., Huang, T., Lai, W. K., and Liu, Q., editors, *Proceedings of the 22nd*

International Conference on Neural Information Processing (ICONIP 2015), pages 148–155, Istanbul, Turkey. Springer International Publishing.

Du, L., Buntine, W., and Jin, H. (2010). A segmented topic model based on the two-parameter Poisson-Dirichlet process. *Machine learning*, 81(1):5–19.

Dumais, S. T., Osuna, E., Platt, J., and Schölkopf, B. (1998). Support Vector Machines. *IEEE INTELLIGENT SYSTEMS*, pages 18–21.

Dy, J. G. and Brodley, C. E. (2004). Feature Selection for Unsupervised Learning. *The Journal of Machine Learning Research*, 5:845–889.

Egozi, O., Gabrilovich, E., and Markovitch, S. (2008). Concept-Based Feature Generation and Selection for Information Retrieval. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI 2008)*, pages 1132–1137, Chicago, Illinois, USA.

Escalante, H. J., García-Limón, M. A., Morales-Reyes, A., Graff, M., Montes-y Gómez, M., Morales, E. F., and Martínez-Carranza, J. (2015). Term-weighting learning via genetic programming for text classification. *Knowledge-Based Systems*, 83:176–189.

Esteban, J., Starr, A., Willetts, R., Hannah, P., and Bryanston-Cross, P. (2005). A Review of data fusion models and architectures: towards engineering guidelines. *Neural Computing and Applications*, 14(4):273–281.

Fan, Y., Guo, J., Lan, Y., Xu, J., Zhai, C., and Cheng, X. (2018). Modeling Diverse Relevance Patterns in Ad-hoc Retrieval. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR 2018)*, pages 375–384, Ann Arbor, MI, USA. ACM.

Fang, H., Tao, T., and Zhai, C. (2004). A Formal Study of Information Retrieval Heuristics. In Sanderson, M., Järvelin, K., Allan, J., and Bruza, P., editors, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2004)*, pages 49–56, Sheffield, UK. ACM, ACM.

Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996). Knowledge Discovery and Data Mining: Towards a Unifying Framework. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 82–88, Portland, Oregon, USA. AAAI Press.

Feldman, R., Aumann, Y., Amir, A., Zilberstein, A., and Klösgen, W. (1997). Maximal Association Rules: A New Tool for Mining for Keyword Co-Occurrences in Document Collections. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pages 167–170, Newport Beach, California, USA. AAAI Press.

Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research*, 3:1289–1305.

Frawley, W. J., Piatetsky-shapiro, G., and Matheus, C. J. (1992). Knowledge Discovery in Databases : An Overview. *AI Magazine*, 13(3):57–70.

Fürnkranz, J. (1998). A Study Using n-gram Features for Text Categorization. Technical Report 1998, Austrian Research Institute for Artifical Intelligence.

Gable, G. G. (1994). Integrating case study and survey research methods: an example in information systems. *European journal of information systems*, 3(2):112–126.

Galliers, R. D. (1992). Choosing appropriate information systems research approaches: a revised taxonomy. In Nissen, H.-E., Klein, H., and Hirschheim, R., editors, *Information Systems Research: Contemporary Approaches & Emergent Traditions*. North Holland, Amsterdam, The Netherlands, 1 edition.

Gao, Y. (2015). *Pattern-based Topic Modelling and its Application for Information Filtering and Information Retrieval*. PhD thesis, Queensland University of Technology.

Gao, Y., Li, Y., Lau, R. Y. K., Xu, Y., and Bashar, M. A. (2017). Finding Semantically Valid and Relevant Topics by Association-Based Topic Selection Model. *ACM Transactions on Intelligent Systems and Technology*, 9(1):1–22.

Gao, Y., Xu, Y., and Li, Y. (2013). Pattern-Based Topic Models for Information Filtering. In *Proceeding of the IEEE 13th International Conference on Data Mining Workshops (ICDM Workshops)*, pages 921–928, TX, USA. IEEE Computer Society.

Gao, Y., Xu, Y., and Li, Y. (2014a). A Topic based Document Relevance Ranking Model. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pages 271–272. International World Wide Web Conferences Steering Committee.

Gao, Y., Xu, Y., and Li, Y. (2014b). Topical Pattern Based Document Modelling and Relevance Ranking. In Benatallah, B., Bestavros, A., Manolopoulos, Y., Vakali, A., and Zhang, Y., editors, *Proceedings of the 15th International Conference in Web Information Systems Engineering (WISE 2014)*, pages 186–201, Thessaloniki, Greece. Springer International Publishing.

Gao, Y., Xu, Y., and Li, Y. (2015). Pattern-based Topics for Document Modelling in Information Filtering. *IEEE Transactions on Knowledge and Data Engineering*, 27(6).

Goutsias, J., Mahler, R. P., and Nguyen, H. T. (1997). *Random Sets: Theory and Applications*. Springer-Verlag New York.

Greiff, W. R. (1998). A Theory of Term Weighting Based on Exploratory Data Analysis. In Croft, W. B., Moffat, A., van Rijsbergen, C. J., Wilkinson, R., and Zobel, J., editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, pages 11–19, Melbourne, Australia. ACM.

Griffiths, T. L. and Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Hammache, A., Boughanem, M., and Ahmed-Ouamer, R. (2014). Combining compound and single terms under language model framework. *Knowledge and Information Systems*, 39(2):329–349.

Han, J., Cheng, H., Xin, D., and Yan, X. (2007). Frequent pattern mining: Current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86.

Han, J., Pei, J., and Yin, Y. (2000). Mining Frequent Patterns without Candidate Generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD'00)*, pages 1–12, Dallas, Texas, USA. ACM.

He, B., Huang, J. X., and Zhou, X. (2011). Modeling term proximity for probabilistic information retrieval models. *Information Sciences*, 181(14):3017–3031.

He, J., Hu, Z., Berg-Kirkpatrick, T., Huang, Y., and Xing, E. P. (2017). Efficient Correlated Topic Modeling with Topic Embedding. In *Proceedings of the 23rd ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining (KDD '17), pages 225–233, New York, New York, USA. ACM Press.

Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine learning*, 42(1-2):177–196.

Hou, Y., Zhang, P., Yan, T., Li, W., and Song, D. (2010). Beyond redundancies: A metric-invariant method for unsupervised feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 22(3):348–364.

Huang, A. (2008). Similarity Mesaures for Text Document Clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008)*, pages 51–56, Christchurch, New Zealand.

Huang, Y.-F. and Lin, S.-Y. (2003). Mining Sequential Patterns Using Graph Search Techniques. In *Proceedings of the 27th International Computer Software and Applications Conference (COMPSAC 2003): Design and Assessment of Trustworthy Software-Based Systems*, pages 4–9, Dallas, TX, USA. IEEE Computer Society.

Huston, S. and Croft, W. B. (2014). A Comparison of Retrieval Models using Term Dependencies. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM '14)*, pages 111–120, New York, New York, USA. ACM Press.

Hwang, S. J. and Sigal, L. (2014). A Unified Semantic Embedding: Relating Taxonomies and Attributes. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014 (NIPS 2014)*, pages 271–279, Montreal, Quebec, Canada.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8):651–666.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated Gain-Based Evaluation of IR Techniques. *ACM Transactions on Information Systems*, 20(4):422–446.

Jeng, R.-H. and Chen, W.-S. (2016). Two Feature-Level Fusion Methods with Feature Scaling and Hashing for Multimodal Biometrics. *IETE Technical Review*, 34(1):1–11.

Jian, L., Li, J., Shu, K., and Liu, H. (2016). Multi-Label Informed Feature Selection. In Kambhampati, S., editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016)*, pages 1627–1633, New York, NY, USA. IJCAI/AAAI Press.

Joachims, T. (1996). A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Technical report, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In Nedellec, C. and Rouveirol, C., editors, *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, pages 137–142, Chemnitz, Germany. Springer.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, pages 133–142, Edmonton, Alberta, Canada. ACM.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. In Cohen, W. W. and Hirsh, H., editors, *Proceedings of the 11th International Conference on Machine Learning (ICML 1994)*, pages 121–129, New Brunswick, NJ, USA. Morgan Kaufmann.

John Lu, Z. Q. (2010). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, volume 173.

Jones, K. S., Walker, S., and Robertson, S. E. (2000a). A probabilistic model of information retrieval: development and comparative experiments: Part 1. *Information Processing & Management*, 36(6):779–808.

Jones, K. S., Walker, S., and Robertson, S. E. (2000b). A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information Processing & Management*, 36(6):809–840.

Kaszkiel, M. and Zobel, J. (1997). Passage Retrieval Revisited. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, pages 178–185, Philadelphia, PA, USA. ACM.

Khan, A., Baharudin, B., Lee, L. H., and Khan, K. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. *Journal of advances in information technology*, 1(1):4–20.

Kim, H. D., Zhai, C., and Han, J. (2010). Aggregation of Multiple Judgments for Evaluating Ordered Lists. In Gurrin, C., He, Y., Kazai, G., Kruschwitz, U., Little, S., Roelleke, T., Rüger, S. M., and van Rijsbergen, K., editors, *Proceedings of the 32nd European Conference on IR Research (ECIR 2010)*, pages 166–178, Milton Keynes, UK. Springer.

Kludas, J. (2011). *Information fusion for multimedia: exploiting feature interactions for semantic feature selection and construction*. PhD thesis, University of Geneva.

Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324.

Kozorovitsky, A. K. and Kurland, O. (2011a). Cluster-Based Fusion of Retrieved Lists. In *Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, pages 893–902, Beijing, China.

Kozorovitsky, A. K. and Kurland, O. (2011b). From “Identical” to “Similar”: Fusing Retrieved Lists Based on Inter-Document Similarities. *Journal of Artificial Intelligence Research*, 41:267–296.

Krikon, E. and Kurland, O. (2011). A study of the integration of passage-, document-, and cluster-based information for re-ranking search results. *Information Retrieval*, 14(6):593–616.

Kruse, R., Schwecke, E., and Heinsohn, J. (1991). *Uncertainty and Vagueness in Knowledge Based Systems: Numerical Methods*. Springer-Verlag Berlin Heidelberg.

Lavrenko, V. and Croft, W. B. (2001). Relevance-Based Language Models. In Croft, W. B., Harper, D. J., Kraft, D. H., and Zobel, J., editors, *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR’01)*, pages 120–127, New Orleans, Louisiana, USA. ACM.

Lee, C.-J., Ai, Q., Croft, W. B., and Sheldon, D. (2015a). An Optimization Framework for Merging Multiple Result Lists. In Bailey, J., Moffat, A., Aggarwal, C. C., de Rijke, M., Kumar, R., Murdock, V., Sellis, T. K., and Yu, J. X., editors, *Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM 2015)*, Melbourne, VIC, Australia. ACM.

Lee, Y.-H., Hu, P. J.-H., and Tu, C.-Y. (2015b). Ontology-Based Mapping for Automated Document Management: A Concept-Based Technique for Word Mismatch and Ambiguity Problems in Document Clustering. *ACM Transactions on Management Information Systems (TMIS)*, 6(1):4.

Leedy, P. D. and Ormrod, J. E. (2005). *Practical research: Planning and design*. Prentice Hall, Upper Saddle River, N.J, 8 edition.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. (2004). RCV1: A New Benchmark Collection for Text Categorization Research. *The Journal of Machine Learning Research*, 5:361–397.

Li, B., Wang, B., Zhou, R., Yang, X., and Liu, C. (2016). CITPM: A Cluster-Based Iterative Topical Phrase Mining Framework. In Navathe, B. S., Wu, W., Shekhar, S., Du, X., Wang, S. X., and Xiong, H., editors, *Database Systems for Advanced Applications: 21st International Conference, DASFAA 2016, Dallas, TX, USA, April 16-19, 2016, Proceedings, Part I*, pages 197–213, Cham. Springer International Publishing.

Li, C. H., Song, W., and Park, S. C. (2009a). An automatically constructed thesaurus for neural network based document categorization. *Expert Systems with Applications*, 36(8):10969–10975.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2017a). Feature Selection: A Data Perspective. *ACM Computing Surveys*, 50(6).

Li, W. and McCallum, A. (2006). Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584. ACM.

Li, X., Gao, J., Li, H., Yang, L., and Srihari, R. K. (2013). A Multimodal Framework for Unsupervised Feature Fusion. In He, Q., Iyengar, A., Nejdl, W., Pei, J., and Rastogi, R., editors, *Proceeding of the 22nd ACM International Conference on Information and Knowledge Management (CIKM'13)*, pages 897–902, San Francisco, CA, USA.

Li, Y. (2003). Extended Random Sets for Knowledge Discovery in Information Systems. In Wang, G., Liu, Q., Yao, Y., and Skowron, A., editors, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 9th International Conference, RSFDGrC 2003, Chongqing, China, May 26–29, 2003 Proceedings*, pages 524–532. Springer Berlin Heidelberg, Berlin, Heidelberg.

Li, Y., Algarni, A., Albathan, M., Shen, Y., and Bijaksana, M. A. (2015). Relevance Feature Discovery for Text Mining. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1656–1669.

Li, Y., Algarni, A., Wu, S.-T., and Xue, Y. (2009b). Mining Negative Relevance Feedback for Information Filtering. In *Proceedings of the 2009 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2009)*, pages 606–613, Milan, Italy. IEEE Computer Society.

Li, Y., Algarni, A., and Xu, Y. (2011). A pattern mining approach for information filtering systems. *Information Retrieval*, 14(3):237–256.

Li, Y., Algarni, A., and Zhong, N. (2010). Mining Positive and Negative Patterns for Relevance Feature Discovery. In Rao, B., Krishnapuram, B., Tomkins, A., and Yang, Q., editors, *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10)*, pages 753–762, Washington, DC, USA. ACM.

Li, Y., Li, T., and Liu, H. (2017b). Recent advances in feature selection and its applications. *Knowledge and Information Systems*, 53(3):551–577.

Li, Y., Wu, S.-T., and Xu, Y. (2004). Deploying Association Rules on Hypothesis Spaces. In *Proceedings of the International Conference on Computioonal Intelligence for Modelling, Control and Automation (CIMCA 2004)*, Gold Coast, QLD, Australia. University of Canberra.

Li, Y. and Yao, Y. Y. (2002a). User Profile Model: A View from Artificial Intelligence. In *Rough Sets and Current Trends in Computing*, volume 2475 of *Lecture Notes in Computer Science*, pages 493–496. Springer.

Li, Y. and Yao, Y. Y. (2002b). User Profile Model: A View from Artificial Intelligence. In Alpigini, J. J., Peters, J. F., Skowron, A., and Zhong, N., editors, *Rough Sets and Current Trends in Computing: Third International Conference, RSCTC 2002 Malvern, PA, USA,* *October 14–16, 2002 Proceedings*, pages 493–496. Springer Berlin Heidelberg, Berlin, Heidelberg.

Li, Y., Zhang, L., Xu, Y., Yao, Y., Lau, R., and Wu, Y. (2017c). Enhancing Binary Classification by Modeling Uncertain Boundary in Three-Way Decisions. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):1438–1451.

Li, Y. and Zhong, N. (2003). Interpretations of association rules by granular computing. In *Third IEEE International Conference on Data Mining*, pages 593–596. IEEE Comput. Soc.

Li, Y. and Zhong, N. (2004). Web mining model and its applications for information gathering. *Knowledge-Based Systems*, 17(5-6):207–217.

Li, Y., Zhong, N., and Yao, Y. Y. (2005). Topic-oriented mining and reasoning. In *Proceedings of the 2005 International Conference on Active Media Technology (AMT 2005).*, pages 321–326. IEEE.

Li, Y., Zhou, X., Bruza, P., Xu, Y., and Lau, R. Y. K. (2008). A Two-stage Text Mining Model for Information Filtering. In Shanahan, J. G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D. A., Kolcz, A., Choi, K.-S., and Chowdhury, A., editors, *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*, pages 1023–1032, Napa Valley, California, USA. ACM.

Li, Y., Zhou, X., Bruza, P., Xu, Y., and Lau, R. Y. K. (2012). A two-stage decision model for information filtering. *Decision Support Systems*, 52(3):706–716.

Lillis, D., Toolan, F., Collier, R., and Dunnion, J. (2006). ProbFuse: A Probabilistic Approach to Data Fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'06)*, pages 139–146, Seattle, Washington, USA.

Lillis, D., Toolan, F., Collier, R., and Dunnion, J. (2008). Extending Probabilistic Data Fusion Using Sliding Windows. In *Proceedings of the 30th European Conference on IR Research (ECIR 2008)*, pages 358–369, Glasgow, UK.

Lillis, D., Zhang, L., Toolan, F., Collier, R. W., Leonard, D., and Dunnion, J. (2010). Estimating Probabilities for Effective Data Fusion. In *Proceeding of the 33rd International ACM SIGIR*

Conference on Research and Development in Information Retrieval (SIGIR 2010), pages 347–354, Geneva, Switzerland.

Liu, H., Dougherty, E., Dy, J. G., Torkkola, K., Tuv, E., Peng, H., Ding, C., Long, F., Berens, M., and Parsons, L. (2005). Evolving Feature Selection. *IEEE Intelligent Systems*, 20(6):64–76.

Liu, H. and Yu, L. (2005). Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502.

Liu, J., Ren, X., Shang, J., Cassidy, T., Voss, C. R., and Han, J. (2016). Representing Documents via Latent Keyphrase Inference. In *WWW '16*, pages 1057–1067, New York, New York, USA. ACM Press.

Liu, T., Liu, S., Chen, Z., and Ma, W.-Y. (2003). An Evaluation on Feature Selection for Text Clustering. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML'03)*, pages 488–495, Washington, DC, USA. AAAI Press.

Liu, X. and Croft, W. B. (2002). Passage Retrieval Based on Language Models. In *Proceedings of the 2002 ACM CIKM International Conference on Information and Knowledge Management (CIKM'02)*, pages 375–382, McLean, VA, USA. ACM.

Liu, X. and Croft, W. B. (2004). Cluster-Based Retrieval Using Language Models. In Sanderson, M., Järvelin, K., Allan, J., and Bruza, P., editors, *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'04)*, pages 186–193, Sheffield, UK. ACM.

Liu, Y., Loh, H. T., and Sun, A. (2009). Imbalanced text classification: A term weighting approach. *Expert Systems with Applications*, 36:690–701.

Luo, Q., Chen, E., and Xiong, H. (2011). A semantic term weighting scheme for text categorization. *Expert Systems with Applications*, 38(10):12708–12716.

Lv, Y. and Zhai, C. (2009). Positional Language Models for Information Retrieval. In James Allan, Aslam, J. A., Sanderson, M., Zhai, C., and Zobel, J., editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 299–306, Boston, MA, USA. ACM.

Lv, Y. and Zhai, C. (2010). Positional Relevance Model for Pseudo-Relevance Feedback. In *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2010)*, pages 579–586, Geneva, Switzerland.

Ma, T., Zhao, Y., Zhou, H., Tian, Y., Al-Dhelaan, A., and Al-Rodhaan, M. (2019). Natural disaster topic extraction in Sina microblogging based on graph analysis. *Expert Systems with Applications*, 115:346–355.

Macdonald, C. and Ounis, I. (2010). Global Statistics in Proximity Weighting Models. In *SIGIR 2010 WEB N-GRAM Workshop*.

Man, L., Tan, C. L., Jian, S., and Yue, L. (2009). Supervised and Traditional Term Weighting Methods for Automatic Text Categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):721–735.

Manning, C. D., Raghavan, P., and Schütze, H. (2008a). Evaluation in information retrieval. In *Introduction to Information Retrieval*, volume 1. Cambridge university press Cambridge.

Manning, C. D., Raghavan, P., and Schütze, H. (2008b). *Introduction to Information Retrieval*. Cambridge university press Cambridge.

Maxwell, K. T. and Croft, W. B. (2013). Compact Query Term Selection Using Topically Related Text. In Jones, G. J. F., Sheridan, P., Kelly, D., de Rijke, M., and Sakai, T., editors, *Proceedings of the 36th International ACM SIGIR conference on research and development in Information Retrieval (SIGIR'13)*, pages 583–592, Dublin, Ireland. ACM.

McCallum, A. K. (2002). MALLET: A Machine Learning for Language Toolkit.

Metzler, D. and Croft, W. B. (2005). A Markov Random Field Model for Term Dependencies. In Baeza-Yates, R. A., Ziviani, N., Marchionini, G., Moffat, A., and Tait, J., editors, *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'05)*, pages 472–479, Salvador, Brazil. ACM.

Metzler, D. A. (2007). Automatic feature selection in the markov random field model for information retrieval. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 253–262, Lisbon, Portugal. ACM.

Miao, J., Huang, J. X., and Ye, Z. (2012). Proximity-based Rocchio's Model for Pseudo Relevance Feedback. In Hersh, W. R., Callan, J., Maarek, Y., and Sanderson, M., editors, *Proceedings of the 35th International ACM SIGIR conference on research and development in Information Retrieval (SIGIR'12)*, pages 535–544, Portland, OR, USA. ACM.

Mimno, D., Li, W., and McCallum, A. (2007). Mixtures of hierarchical topics with pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, pages 633–640. ACM.

Mimno, D., Wallach, H. M., Talley, E., Leenders, M., and McCallum, A. (2011). Optimizing Semantic Coherence in Topic Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 262–272. Association for Computational Linguistics.

Molchanov, I. (2005). *Theory of Random Sets*. Springer-Verlag London.

Mooney, C. H. and Roddick, J. F. (2013). Sequential Pattern Mining – Approaches and Algorithms. *ACM Computing Surveys*, 45(2):1–39.

Moschitti, A. and Basili, R. (2004). Complex linguistic features for text classification: A comprehensive study. *Proceedings of the 26th European Conference on IR Research*, pages 181–196.

Nguyen, H. T. (2008). Random sets. *Scholarpedia*, 3(7):3383.

Nuray, R. and Can, F. (2006). Automatic ranking of information retrieval systems using data fusion. *Information Processing and Management*, 42:595–614.

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., and Hsu, M. (2001). PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. In *ICDE'01*, pages 215–224. {IEEE} Computer Society.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238.

Petinot, Y., McKeown, K., and Thadani, K. (2011). A hierarchical model of web summaries. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:*

Human Language Technologies: short papers-Volume 2, pages 670–675. Association for Computational Linguistics.

Pickens, J. and Golovchinsky, G. (2008). Ranked Feature Fusion Models for Ad Hoc Retrieval. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM 2008)*, pages 893–900, Napa Valley, California, USA. ACM.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Qiang, J.-P., Chen, P., Ding, W., Xie, F., and Wu, X. (2016). Multi-document summarization using closed patterns. *Knowledge-Based Systems*, 99:28–38.

Ramage, D., Manning, C. D., and Dumais, S. (2011). Partially Labeled Topic Models for Interpretable Text Mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*, pages 457–465, San Diego, California, USA. ACM.

Rasmussen, M. and Karypis, G. (2004). gCLUTO: An Interactive Clustering, Visualization, and Analysis System. Technical Report 7, University of Minnesota,Department of Computer Science and Engineering, Minneapolis, MN.

Robertson, S. and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Robertson, S. E. and Soboroff, I. (2002). The TREC 2002 Filtering Track Report. In Voorhees, E. M. and Buckland, L. P., editors, *Proceedings of The Eleventh Text REtrieval Conference (TREC 2002)*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology (NIST).

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In *The SMART retrieval system: Experiments in Automatic Document Processing*, pages 313–323. Prentice-Hall Inc.

Sabbah, T., Selamat, A., Selamat, M. H., Al-Anzi, F. S., Viedma, E. H., Krejcar, O., and Fujita, H. (2017). Modified Frequency-Based Term Weighting Schemes for Text Classification. *Applied Soft Computing Journal*, 58:193–206.

Saif, A., Ab Aziz, M. J., and Omar, N. (2016). Reducing explicit semantic representation vectors using Latent Dirichlet Allocation. *Knowledge-Based Systems*, 100:145–159.

Salton, G. and Buckley, C. (1988). Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523.

Salton, G., Wong, A., and Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Savaresi, S. M. and Boley, D. L. (2001). On the Performance of Bisecting K-Means and PDDP. In *Proceedings of the First SIAM International Conference on Data Mining (SDM 2001)*, pages 1–14, Chicago, IL, USA. SIAM.

Scott, S. and Matwin, S. (1999). Feature Engineering for Text Classification. In *Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999)*, pages 379–388, Bled, Slovenia. Morgan Kaufmann.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, 34(1):1–47.

Seno, M. and Karypis, G. (2002). SLPMiner: An Algorithm for Finding Frequent Sequential Patterns Using Length-Decreasing Support Constraint. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 418–425, Maebashi City, Japan. IEEE Computer Society.

Shehata, S., Karray, F., and Kamel, M. (2007). A Concept-based Model for Enhancing Text Categorization. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 629–637, San Jose, California, USA. ACM.

Shehata, S., Karray, F., and Kamel, M. S. (2010). An Efficient Concept-Based Mining Model for Enhancing Text Clustering. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1360–1371.

Shen, Y., Li, Y., and Xu, Y. (2012a). Adopting Relevance Feature to Learn Personalized Ontologies. In Thielscher, M. and Zhang, D., editors, *AI 2012: Advances in Artificial Intelligence - Proceedings of the 25th Australasian Joint Conference on Artificial Intelligence*, pages 457–468, Sydney, Australia. Springer, Berlin, Heidelberg.

Shen, Y., Li, Y., Xu, Y., and Tao, X. (2012b). Matching Relevance Features with Ontological Concepts. In *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences* *on Web Intelligence and Intelligent Agent Technology-Volume 03*, pages 190–194. IEEE Computer Society.

Shi, L. and Nie, J.-Y. (2009). Integrating Phrase Inseparability in Phrase-Based Model. In Allan, J., Aslam, J. A., Sanderson, M., Zhai, C., and Zobel, J., editors, *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'09)*, pages 708–709, Boston, MA, USA. ACM.

Shirakawa, M., Hara, T., and Nishio, S. (2015). N-gram IDF: A Global Term Weighting Scheme Based on Information Distance. In *Proceedings of the 24th International Conference on World Wide Web (WWW 2015)*, pages 960–970, Florence, Italy. ACM.

Smucker, M. D., Allan, J., and Carterette, B. (2007). A Comparison of Statistical Significance Tests for Information Retrieval Evaluation. In Silva, M. J., Laender, A. H. F., Baeza-Yates, R. A., McGuinness, D. L., Olstad, B., Olsen, Ø. H., and Falcão, A. O., editors, *Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management (CIKM'07)*, pages 623–632, Lisbon, Portugal.

Snoek, C. G. M., Worring, M., and Smeulders, A. W. M. (2005). Early versus late fusion in semantic video analysis. In Zhang, H., Chua, T.-S., Steinmetz, R., Kankanhall, M. S., and Wilcox, L., editors, *Proceedings of the 13th annual ACM international conference on Multimedia - MULTIMEDIA '05*, page 399, New York, New York, USA. ACM Press.

Soboroff, I. and Robertson, S. (2003). Building a Filtering Test Collection for TREC 2002. In Clarke, C. L. A., Cormack, G. V., Callan, J., Hawking, D., and Smeaton, A. F., editors, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03)*, pages 243–250, Toronto, Canada. ACM.

Soleimani, H. and Miller, D. J. (2016). Semi-supervised Multi-Label Topic Models for Document Classification and Sentence Labeling. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 105–114, Indianapolis, Indiana, USA. ACM.

Somekh, B. (2005). *Action Research: A Methodology For Change And Development: a methodology for change and development*. McGraw-Hill Education (UK), 1 edition.

Song, Q., Ni, J., and Wang, G. (2013). A Fast Clustering-Based Feature Subset Selection Algorithm for High-Dimensional Data. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):1–14.

Steinbach, M., Karypis, G., and Kumar, V. (2000). A Comparison of Document Clustering Techniques. *KDD workshop on text mining*, 400(X).

Steyvers, M. and Griffiths, T. (2007). Probabilistic Topic Models. In Landauer, T., McNamara, D., Dennis, S., and Kintsch, W., editors, *Handbook of Latent Semantic Analysis*, pages 424–440. Laurence Erlbaum, 1 edition.

Tang, B., Kay, S., and He, H. (2016). Toward Optimal Feature Selection in Naive Bayes for Text Categorization. *IEEE Transactions on Knowledge and Data Engineering*, 28(9):2508–2521.

Tao, X. (2009). *Personalised Ontology Learning and Mining for Web Information Gathering*. PhD thesis, QUT.

Tao, X., Li, Y., and Zhong, N. (2011). A Personalized Ontology Model for Web Information Gathering. *IEEE Transactions on Knowledge and Data Engineering*, 23(4):496–511.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288.

Towell, G., Voorhees, E. M., Gupta, N. K., and Johnson-Laird, B. (1995). Learning Collection Fusion Strategies for Information Retrieval. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML 1995)*, pages 540–548, Tahoe City, California, USA.

Turney, P. D. and Pantel, P. (2010). From Frequency to Meaning: Vector Space Models of Semantics. *Journal of artificial intelligence research*, 37:141–188.

Urbano, J., Marrero, M., and Martín, D. (2013). A Comparison of the Optimality of Statistical Significance Tests for Information Retrieval Evaluation. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'13)*, pages 925–928, Dublin, Ireland.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. In *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML 2006)*, pages 977–984, Pittsburgh, Pennsylvania, USA. ACM.

Wang, C., Bi, K., Hu, Y., Li, H., and Cao, G. (2012). Extracting Search-Focused Key N-Grams for Relevance Ranking in Web Search. In Adar, E., Teevan, J., Agichtein, E., and Maarek, Y., editors, *Proceedings of the Fifth International Conference on Web Search and Web Data Mining (WSDM'12)*, pages 343–352, Seattle, WA, USA. ACM.

Wang, C., Song, Y., El-Kishky, A., Roth, D., Zhang, M., and Han, J. (2015). Incorporating World Knowledge to Document Clustering via Heterogeneous Information Networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15)*, pages 1215–1224, Sydney, NSW, Australia. ACM.

Wang, J., Wei, J.-M., Yang, Z., and Wang, S.-Q. (2017). Feature Selection by Maximizing Independent Classification Information. *IEEE Transactions on Knowledge and Data Engineering*, 29(4):828–841.

Wang, S., Zhu, E., Hu, J., Li, M., Zhao, K., Hu, N., and Liu, X. (2019). Efficient Multiple Kernel k-Means Clustering With Late Fusion. *IEEE Access*, 7:61109–61120.

Wang, X., McCallum, A., and Wei, X. (2007). Topical N-grams: Phrase and Topic Discovery, with an Application to Information Retrieval. In *The Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 697–702, Omaha, NE, USA. IEEE.

Wei, X. and Croft, W. B. (2006). LDA-Based Document Models for Ad-hoc Retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'06)*, pages 178–185, Seattle, Washington, USA. ACM.

Weninger, T., Bisk, Y., and Han, J. (2012). Document-Topic Hierarchies from Document Graphs. In *Proceedings of the 221st ACM International Conference on Information and Knowledge Management (CIKM'12)*, pages 635–644, Maui, HI, USA. ACM.

Wu, H. and Gu, X. (2017). Balancing Between Over-Weighting and Under-Weighting in Supervised Term Weighting. *Information Processing and Management*, 53:547–557.

Wu, S., Li, J., Zeng, X., and Bi, Y. (2014). Adaptive Data Fusion Methods in Information Retrieval. *Journal of the Association for Information Science and Technology*, 65(10):2048–2061.

Wu, S. and Mcclean, S. (2006). Performance prediction of data fusion for information retrieval. *Information Processing and Management*, 42(4):899–915.

Wu, S.-T. (2007). *Knowledge Discovery Using Pattern Taxonomy Model in Text Mining*. PhD thesis, Queensland University of Technology, Brisbane.

Wu, S.-T., Li, Y., and Xu, Y. (2006). Deploying Approaches for Pattern Refinement in Text Mining. In *Proceedings of the Sixth International Conference on Data Mining (ICDM'06)*, pages 1157–1161. IEEE.

Wu, S.-T., Li, Y., Xu, Y., Pham, B., and Chen, P. (2004). Automatic Pattern-Taxonomy Extraction for Web Mining. In *Proceeding of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004)*, pages 242–248, Beijing, China. IEEE Computer Society.

Wu, Y., Li, Y., and Xu, Y. (2019). Dual pattern-enhanced representations model for query-focused multi-document summarisation. *Knowledge-Based Systems*, 163:736–748.

Wu, Y., Li, Y., Xu, Y., and Huang, W. (2016). Mining Topically Coherent Patterns for Unsupervised Extractive Multi-document Summarization. In *2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI'16)*, pages 129–136, Omaha, NE, USA. IEEE Computer Society.

Wu, Z., Cai, L., and Meng, H. (2005). Multi-level Fusion of Audio and Visual Features for Speaker Identification. In Zhang, D. and Jain, A. K., editors, *Advances in Biometrics, Proceedings of the 2006 International Conference on Biometrics (ICB 2006)*, pages 493–499, Hong Kong, China. Springer.

Xi, W., Xu-Rong, R., Khoo, C. S. G., and Lim, E.-P. (2001). Incorporating window-based passage-level evidence in document retrieval. *J. Information Science*, 27(2):73–80.

Xiao, Z., de Silva, T. N., Wei, C., Mao, K., and Ng, G. W. (2016). Constructing Bayesian Networks by Harvesting Knowledge from Online Resources. In *Proceedings of the 19th International Conference on Information Fusion (FUSION 2016)*, pages 106–113, Heidelberg, Germany. IEEE.

Xiong, D., Zhang, M., and Wang, X. (2015). Topic-Based Coherence Modeling for Statistical Machine Translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):483–493.

Xu, C., Wu, Y., and Liu, Z. (2017). Multimodal Fusion with Global and Local Features for Text Classification. In Liu, D., Xie, S., Li, Y., Zhao, D., and El-Alfy, E.-S. M., editors, *Proceedings of the 24th International Conference on Neural Information Processing (ICONIP 2017)*, pages 124–134, Guangzhou, China. Springer.

Xu, G., Wu, X., Yao, H., Li, F., and Yu, Z. (2019). Research on Topic Recognition of Network Sensitive Information Based on SW-LDA Model. *IEEE Access*, 7:21527–21538.

Xu, Y., Li, Y., and Shaw, G. (2011). Reliable representations for association rules. *Data & Knowledge Engineering*, 70(6):555–575.

Xue, X., Huston, S., and Croft, W. B. (2010). Improving Verbose Queries using Subset Distribution. In *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM 2010)*, pages 1059–1068, Toronto, Ontario, Canada. ACM.

Xue, X.-B. and Zhou, Z.-H. (2009). Distributional Features for Text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 21(3):428–442.

Yan, X., Cheng, H., Han, J., and Xin, D. (2005). Summarizing Itemset Patterns: A Profile-Based Approach. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*, pages 314–323, Chicago, Illinois, USA. ACM.

Yang, Y. and Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97)*, pages 412–420, Nashville, TN, USA. Morgan Kaufmann Publishers Inc.

Yao, Y. (2009). Three-Way Decision: An Interpretation of Rules in Rough Set Theory. In Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., and Wang, G., editors, *Rough Sets and Knowledge Technology, Proceeding of the 4th International Conference on Rough Sets and Knowledge Technology (RSKT 2009)*, pages 642–649, Gold Coast, Australia. Springer.

Yao, Y. Y. (2001). On Modeling Data Mining with Granular Computing. In *Proceeding of the 25th Annual International Computer Software and Applications Conference (COMPSAC 2001)*, pages 638–643, Chicago, IL, USA. IEEE, IEEE Computer Society.

Yi, X. and Allan, J. (2009). A Comparative Study of Utilizing Topic Models for Information Retrieval. In Boughanem, M., Berrut, C., Mothe, J., and Soule-Dupuy, C., editors, *Advances in Information Retrieval, Proceeding of the 31th European Conference on IR Research (ECIR 2009)*, pages 29–41, Toulouse, France. Springer Berlin Heidelberg.

Yin, J. and Wang, J. (2014). A Dirichlet Multinomial Mixture Model-based Approach for Short Text Clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14)*, pages 233–242, New York, New York, USA. ACM Press.

Yu, S., Tranchevent, L., Liu, X., Glanzel, W., Suykens, J. A. K., De Moor, B., and Moreau, Y. (2011). Optimized Data Fusion for Kernel k-Means Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):1031–1039.

Yuefeng, L. and Ning, Z. (2006). Mining Ontology for Automatically Acquiring Web User Information Needs. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):554–568.

Zaki, M. J. (2001). SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine Learning*, 42(1/2):31–60.

Zhai, C. and Lafferty, J. (2004). A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214.

Zhang, J. D. and Chow, C. Y. (2016). CRATS: An LDA-Based Model for Jointly Mining Latent Communities, Regions, Activities, Topics, and Sentiments from Geosocial Network Data. *IEEE Transactions on Knowledge and Data Engineering*, 28(11):2895–2909.

Zhang, S. and Balog, K. (2017). Design Patterns for Fusion-Based Object Retrieval. In Jose, J. M., Hauff, C., Altingövde, I. S., Song, D., Albakour, D., Watt, S. N. K., and Tait, J., editors, *Advances in Information Retrieval - 39th European Conference on IR Research (ECIR 2017)*, pages 684–690, Aberdeen, UK. Springer International Publishing.

Zhang, Z., Wang, Q., Si13, L., and Gao, J. (2016). Learning for Efficient Supervised Query Expansion via Two-stage Feature Selection. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016)*, pages 265–274, Pisa, Italy.

Zhao, Z. and Liu, H. (2007). Spectral Feature Selection for Supervised and Unsupervised Learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML'07)*, pages 1151–1157, Corvalis, Oregon, USA. ACM.

Zhao, Z., Wang, L., Liu, H., and Ye, J. (2013). On Similarity Preserving Feature Selection. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):619–632.

Zheng, Z., Wu, X., and Srihari, R. (2004). Feature Selection for Text Categorization on Imbalanced Data. *SIGKDD Explorations*, 6(1):80–89.

Zhong, N., Li, Y., and Wu, S.-T. (2012). Effective Pattern Discovery for Text Mining. *IEEE Transactions on Knowledge and Data Engineering*, 24(1):30–44.

Zhou, D., Lawless, S., Min, J., and Wade, V. (2010). A Late Fusion Approach to Cross-lingual Document Re-ranking. In Huang, J., Koudas, N., Jones, G. J. F., Xindong Wu, K. C.-T., and An, A., editors, *Proceedings of the 19th ACM Conference on Information and Knowledge Management (CIKM'10)*, pages 1433–1436, Toronto, Ontario, Canada. ACM.

Zhou, X., Li, Y., Bruza, P., Xu, Y., and Lau, R. Y. K. (2008). Pattern Taxonomy Mining for Information Filtering. In Wobcke, W. and Zhang, M., editors, *AI 2008: Advances in Artificial Intelligence: 21st Australasian Joint Conference on Artificial Intelligence Auckland, New Zealand, December 1-5, 2008. Proceedings*, pages 416–422, Berlin, Heidelberg. Springer Berlin Heidelberg.

Zhou, X., Li, Y., Bruza, P., Xu, Y., and Lau, R. Y. K. (2011). Pattern Mining for a Two-Stage Information Filtering System. In Huang, J. Z., Cao, L., and Srivastava, J., editors, *Advances in Knowledge Discovery and Data Mining (PAKDD 2011)*, pages 363–374, Shenzhen, China. Springer Berlin Heidelberg.

Zhu, G. and Iglesias, C. A. (2017). Computing Semantic Similarity of Concepts in Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):72–85.

Zhu, J., Ahmed, A., and Xing, E. P. (2012). MedLDA: Maximum Margin Supervised Topic Models. *Journal of Machine Learning Research*, 13:2237–2278.

Zhu, W. and Lin, Y. (2013). Using Gini-index for Feature Weighting in Text Categorization. *Journal of Computational Information Systems*, 9(14):5819–5826.