Comparing XML-IR Query Formation Interfaces

Alan Woodley[1], *Shlomo Geva*[1], *Sylvia Lauretta Edwards*[2]

[1]School of Software Engineering and Data Communications
[2]School of Information Systems
Faculty of Information Technology
Queensland University of Technology
QLD 4000 Australia

{*a.woodley,s.geva,s.edwards*}*@qut.edu.au*

Abstract *XML information retrieval (XML-IR) systems differ from traditional information retrieval systems by using structure of XML documents to retrieve more specific units of information than the documents themselves. Users interact with XML-IR systems via structured queries that express their content and structural requirements. Historically, it has been common belief within the XML-IR community that structured queries will perform better than traditional keyword-only queries. However, recent system-orientated analysis has show that this assumption may be incorrect when system performance is averaged over a set of queries. Here, we test this assumption with users via a simulated work task experiment. We compare a keyword only interface with two user friendly XML-IR interfaces: NLPX, a natural language interface and Bricks, a query-by-template interface. This is the first time that a XML-IR natural language interface has been tested in user experiments. We compare the retrieval performance of all three interfaces and the usability of the two structured interfaces. Our results correspond to those of the system-orientated evaluation and indicate that structured queries do not aid retrieval performance. They also show that in terms of retrieval performance and usability the structured interfaces are comparable.*

Keywords Users, Information Retrieval, XML.

1 Introduction

Traditional information retrieval (IR) systems respond to user queries with a ranked list of relevant documents. XML documents (Figure 1) explicitly separate content and structure. By incorporating structure into the retrieval process XML Information Retrieval (XML-IR) systems are able to return highly specific information to users, lower than the document level. This has the potential to be highly beneficial to users since it can provide very specific responses to their information needs.

Proceedings of the 11th Australasian Document Computing Symposium, Brisbane, Australia, December 11, 2006.
Copyright for this article remains with the authors.

```
<article>
    <author>Roger Fuller</author>
    <title>Toward a robust Martian-English
    translator</title>
    <section>
        <title>Introduction</title>
        <paragraph>Because of a dramatic
        lack of interpreters, Communication
        between <bold>Martians</bold> and
        <bold>Terrestrians</bold> is confronted
        to...</paragraph>
    </section>
    <section>
        <title>Introduction</title> ...
    </section>
    ...
</article>
```

Figure 1: XML representation of a scientific article.

The INitiative for the Evaluation of XML Retrieval (INEX) [1] is an organisation that was established to facilitate collaboration between XML-IR researchers. INEX is comparable to TREC [10] and provides a test collection consisting of simulated information needs (topics), a collection of XML documents and for each topic a set of relevant XML elements. INEX has differentiated itself from TREC in two ways: first, by returning results lower than the document level (that is XML Elements) and by facilitating the retrieval of two distinct types of topics - Content Only (CO) and Content and Structure (CAS).

The difference between CO and CAS topics is two fold. First, while they both contain users' content requirements, CAS topics also contain users' structural requirements. For this reason CAS topics are also referred to as structured queries. Second, CO topics express users information needs in keywords while CAS topics express users' information needs in formal languages such as NEXI [6]. A typical CAS topic appears in Figure 2 with its *castitle* element containing a NEXI expression. The addition of structure in CAS topics enables users to write more powerful queries since they are able to direct their search to elements within an XML document that best suit

```
<topic topic_id="275" query_type="CAS" ct_no="131">
    <castitle>//article[about(.//abs, "data mining")]//sec[about(., "frequent itemsets")]</castitle>
    <description>sections about frequent itemsets from articles with abstract about data mining</description>
    <narrative>To be relevant, a component has to be a section about "frequent itemsets". For example, it could be
    about algorithms for finding frequent itemsets, or uses of frequent itemsets to generate rules. Also, the article must
    have an abstract about "data mining". I need this information for a paper that I am writing. It is a survey of different
    algorithms for finding frequent itemsets. The paper will also have a section on why we would want to find frequent
    itemsets.</narrative>
</topic>
```

Figure 2: INEX topic 275 that contains a formal language query (castitle), a natural language query (description) and an information need context (narrative).

their information need. A premise of XML-IR is that this additional information will better fulfil users' information needs, although, recent system based evaluation may contradict this premise [8].

We tested this premise by preforming an interactive XML-IR experiment using both keyword and structured interfaces. For our keyword interface we used a traditional IR interface similar to the interfaces used by Internet search engines. However, a keyword only interface is not able to capture the structural needs of users, which is the reason that INEX has used formal languages such as NEXI [6] to capture users' structural and content requirements. However, formal queries language are too difficult to use by expert - let alone casual users; are too tightly bound to the physical structure of the document and do not scale well across heterogeneous collections. Usability testing has validated the problems that casual users have with formulating NEXI queries [9].

Clearly, a formal language interference is unsuitable for non-laboratory XML-IR use; therefore, XML-IR researchers have investigated alternative user-oriented interfaces. Here, we discuss two such interfaces: NLPX, a natural language interface where users enter queries written in English (such as in Figure 2's *description* element) and Bricks, a query-by-template interface where users enter queries via a graphical user interface. Furthermore, we detail and present outcomes from a useability experiment that compared the retrieval performance of both interfaces with a keyword only interface. This is the first time that a natural language interface has been tested with users. Our results indicate that there is little difference in terms of retrieval between all three interfaces or usability between the structured interfaces.

2 Motivation

The motivation for this research is based on a need to investigate the task of query formation in an interactive XML-IR setting, a gap in current knowledge. Our research focus manifests itself in two areas. First, we want to observe if the addition of structural requirements to queries aids in retrieval. Second, we want to investigate alternative means of users formulating structured queries.

2.1 Previous XML-IR Evaluation

The majority of XML-IR system evaluation has been batch testing using a version of the Cranfield Methodology, which uses a controlled set of queries and relevance judgments [3]. Historically, this method of evaluation has been very successful for evaluating and improving the retrieval performance of IR systems and algorithms, particularly in traditional IR [10]. This method allows for repeated and extensive testing of systems within a laboratory setting, however, since the method does not involve actual users there is no way of guaranteeing that their full needs are being met. In fact, research on traditional IR systems has shown that improved retrieval performance in a laboratory environment does not always correlate with user satisfaction [5]. The field of interactive information retrieval evaluation was established to collect quantitative and qualitative feedback from users regarding their use of IR systems. However, most interactive XML-IR experiments [7] have focused on results presentation (for instance: do users prefer to read several paragraphs or one section) rather than query formation. This work presents one of the first investigations into how users formulate queries for use in an XML-IR system. And in particular, the first time that a natural language interface has been compared to other interfaces,

2.2 Adding Structural Hints to Queries

XML-IR system return document fragments (elements) rather than entire documents. A common belief amongst XML-IR researchers has been that adding structural hints to queries will improve retrieval performance, in particular precision. The premise stems from a belief that by adding structural hints users will be able to focus retrieval more closely to elements that match their information need. Historically, this premises has not been verified. Even though INEX has had sperate tracks that deal specifically with content only queries (CO) and content and structure queries (CAS), they have always used different topics; thereby, disabling valid comparison between the two tracks.

In 2005, INEX decided to verify this premise by the introduction of an additional CO+S track [2]. The premises of the CO+S track was for a user to perform

```
<topic topic_id="202" query_type="CO+S" ct_no="1">
    <title>ontologies case study</title>
    <castitle>//article[about(., ontologies)]//sec[about(., ontologies case study)]</castitle>
    <description>Case studies in the use of ontologies</description>
    <narrative>I'm writing a report on the use of ontologies. I'm interested in knowing how ontologies are used to
    encode knowledge in real world scenarios. I'm particularly interested in knowing what sort or concepts and relations
    people use in their ontologies. I'm not interested in general ontology frameworks or technical details about tools for
    ontology creation or management. An example relevant result contains a description of the real world phenomena
    described by the ontology and also lists some of the concepts used and relations between concepts. </narrative>
</topic>
```

Figure 3: INEX topic 202 that contains both a CO query (title) and a CO+S query (castitle)).

a standard IR interaction using an initial content only query, however, if the user was unsatisfied with the results list (for instance it contained too many irrelevant elements) then he/she could narrow down their search by creating a second version that contained a structural hint. CO and CO+S queries were encapsulated within the same topic, and shared the same set of relevance judgements thereby allowing for valid comparison. Figure 3 is an example of a CO/CO+S topic. The CO query is expressed as keywords in the title element, and the CAS query is expressed in formal language in the castitle element.

The retrieval performance of participants in both the CO and CO+S tracks was analysed by Trotman and Lalmas [8]. They showed that while some systems performed better in the CO+S track, none of the improvements were statistically significant. They concluded that the reason that the addition of structure did not improve retrieval performance was because users were not able to write meaningful structured queries. However, they also suggest that this may be a problem of the INEX's source collection. A third alternative, not suggested by the authors, was that current XML-IR systems are not able to process structured queries effectively.

Regardless of the outcomes of Trotman and Lalmas' study, we feel that it is important to observe the effect of adding structure to queries within the field of XML-IR, for several reasons. First, while the addition of structure to XML-IR queries has been investigated in system-orientated (or Cranfield-like) testing, it has not been fully investigated in interactive XML-IR experiments. Secondly, the addition of structure may help retrieval in different collections or if it is better handled by XML-IR systems. It should also be noted that the IR system used as in our experiments performed better using CO+S queries, rather than CO, in batch testing.

2.3 XML-IR Interfaces

There are two standard interfaces for interacting with XML-IR systems, keywords and structured formal languages; however, neither interface optimally addresses the needs of XML-IR users. Here, we discuss the problems associated with both types of interfaces, and outline how they can be solved using alternative interfaces such as natural language or query-by-template. First, keyword based interfaces are too unsophisticated to fully capture XML-IR users' complex information needs since users are unable to specify structural constraints. For instance, in the information need present in Figure 2 the user only wants to search in abstracts and sections, which they are unable to specify just using keywords. Secondly, users may wish to search parts of documents that they do not intend to retrieve, but are rather used to aid (or support) their retrieval. For example, in the information need present in Figure 2 the user wants to retrieve sections from articles that with an abstract on data mining, however, they do not wish to retrieve the abstract itself. Again, this information can not be conveyed just using keywords.

The complexity of XML-IR has lead to the development of formal query languages (akin to SQL for databases) specifically designed for XML-IR, such as NEXI [6]. A sample NEXI expression is presented in the castitle tag in Figure 2. However, formal query languages have also posed problems. First, formal query languages are too difficult for users, both expert and casual, to correctly express their structural and content information needs. Examples of difficulties experienced by expert users occurred at the 2003 and 2004 INEX Workshops where 63 per cent and 12 per cent of queries constructed by experts had major semantic or syntactic errors. It has already been shown that XML-IR users find query-by-template interfaces easier to use than formal language interfaces [9] and users should be able to intuitively express their information need in a natural language. Second, formal query languages are too tightly bound to the physical structure of documents; hence, users need to know the physical tag names of elements in order to express their structural needs. While this information may be obtained from a document's DTD or Schema, users are unlikely to remember hundreds of tags names; furthermore, due to security/privacy reasons, there are situations where the proprietor of the collection does not wish to grant public access to those files. The problem is magnified in heterogeneous collections since a single tag can have multiple names. In contrast, structural requirements in both natural language and

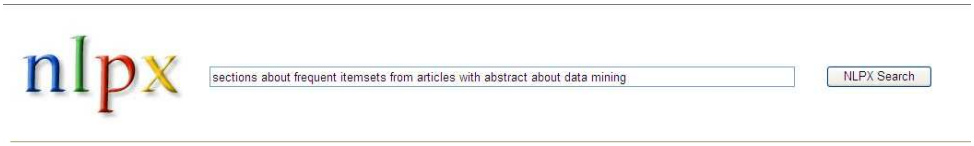

Figure 4: NLPX input using the information need in Figure 2 as a sample query

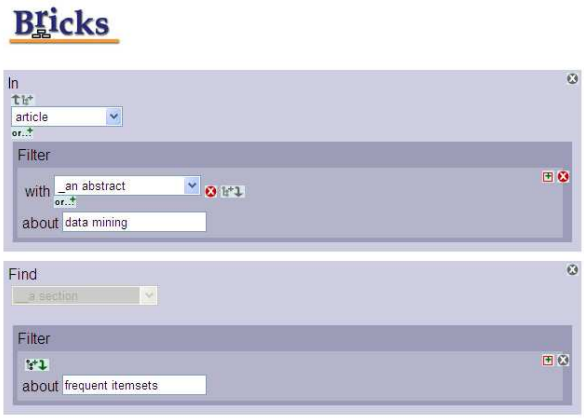

Figure 5: Bricks input using the information need in Figure 2 as a sample query

query-by-templates can be expressed at a higher conceptual level, allowing the underlying documents' structure to be completely hidden from users

3 Experimental System

The system used in the experiment is separated into two parts: the front-end interfaces and the backend retrieval system. Two different interfaces were used: NLPX, that accepted queries written in natural language (English) [11][12], and Bricks, a query by template interface that allowed users to enter queries via a graphical user interface [9]. Examples of the input screen used for both interfaces appear in Figure 4 and Figure 5. These examples capture the information need expressed in the description element of Figure 2 and are a representative of the type of queries entered by the participants. The same backend search engine, GPX, was used for both interfaces. Since GPX only accepted formal language queries, both interfaces translated their user input into NEXI before submitting them to GPX. Below we describe NLPX, Bricks and GPX in more detail.

3.1 Interface A - NLPX

NLPX accepts natural language queries (NLQs) and produces formal queries written in the NEXI language. The NLPX translation process involves four steps. First, NLPX tags words either as special connotations (for instance structures) or by their part of speech. Second NLPX divides sentences into atomic, non overlapping segments (called chunks) and then classifies them into grammatical classes. Third, NLPX matches the tagged NLQs to query templates that were derived from the inspection of previous INEX queries. Finally, NLPX outputs the query in NEXI format. Batch testing of a single backend search engine that used both natural language queries parsed through NLPX and formal NEXI queries has shown comparable results [12]. This is the first time that NLPX has been tested in a usability experiment.

3.2 Interface B - Bricks

Bricks is a query-by-template interface that allows users to input structured queries via a graphical user interface (GUI). Users enter their content needs via text boxes and their structural needs via drop-down boxes. To aid users, structural needs were indicated via conceptual rather than physical names, for example "a section" rather than sec. Bricks allows users to develop queries in several steps ("blocks") starting with their desired unit of retrieval and then by adding any additional information needs. Blocks were also added as the user traversed the hierarchy of the documents (for instance from *article* to *section* to *paragraph*). Upon completion of input, the data in the Bricks GUI was translated to formal NEXI expression, however, due to the constraints of the GUI, users were unable to enter malformed expressions. Usability testing has shown that users find Bricks superior to keyword only and NEXI interfaces [9].

3.3 Backend Retrieval System- GPX

The backend retrieval system for this experiment was Gardens Point X (GPX) [4]. GPX was chosen since it has performed strongly at the annual INEX conference since 2002 - consistently among the top three systems. GPX stores the information about each leaf element in the collection as an inverted list. Upon retrieval, GPX

Table 1: The order of the information needs us by each user group

Sub-Experiment (interface)	1 (NLPX)		2 (NLPX)		3 (Bricks)	
Topic Order	1	2	3	4	5	6
Group A	253	256	257	270	275	284
Group B	275	284	253	256	257	270
Group C	257	270	275	284	253	256

matches query terms to all leaf elements that contain the term and then dynamically creates their ancestors. Elements are ranked according to their predicted relevance in GPX's ranking scheme. GPX rewards leaf elements that contain phrases and specific, rather than common, terms. It also rewards ancestors with multiple relevant children, rather than a single relevant child. For this experiment, the results list was filtered so that "overlapping elements" (that is, elements whose ancestors or descendants appear higher ranked on the results list) were removed before been being presented to users. This decision was made because users have been known to react negatively to overlapping elements [7].

4 Experimental Methodology

4.1 Participants, Collection and Information Requests

The experiment simulated the task of users interacting with an academic retrieval system. Sixteen participants took part in the experiment. The participants acted as academic researchers, for example: post-graduate research students, corporate researchers or academics. The participants searched a collection of academic journal articles, specifically IEEE journal articles from 1995 to 2002. The journals had a broad range of focus, ranging from general journals such as Computing to specific journals such as Neutral Networks (the complete list can be foundd in the annual INEX proceedings [1] [2]).

The participants were post-graduate information technology students who were uninitiated in the domain of XML-IR. While this may not be a representative sample of possible XML-IR users, it was necessary to have such participants since understanding the technical nature of the information needs and source collection was beyond casual users. Also since the participants were uninitiated in the domain of XML-IR, it is valid for us for us to extrapolate the results of this experiment into the wider area of XML-IR.

The participants were given six information needs that simulated those of a real user. The information needs contained both a detailed explanation of the information sought and a condition of relevance that described the motivation behind the information need. The information needs were sampled from the narrative elements of INEX Topics 253 - 284; an example information need was presented in the narrative element of Figure 2.

4.2 Sub-Experiments

The participants preformed three sub-experiments that correlated to three different methods of translating information needs. The first two sub-experiments used the NLPX interface, whereas the last sub-experiment used the Bricks interface. For each sub-experiment, the participants attempted to fulfil two information needs. To reduce bias, the participants where split into three groups and used the information needs in the order presented in Table 1. For each information need the participants interacted with the interfaces by submitting queries and receiving back matching information items, which may or may not be relevant.

The first sub-experiment examined users' initial reaction to using the NLPX interface. They were instructed to enter keyword only queries into NLPX as if it were a standard Internet search engine. Participants were then given a short tutorial about structured information retrieval and were shown some examples of structured natural language queries. The participants then performed the second sub-experiment by entering structured queries into NLPX. A second tutorial was then given on how to use the Bricks interface to preform structured queries. Following this, the participants performed the final sub-experiment were they entered structured queries using the Bricks interface.

Following the experiment, feedback from participants was sought in two ways: first, a survey conducted directly after the experiment and second, one-on-one interviews conducted in the weeks following the experiment that were recorded and later transcribed. During the experiment, the actions of the participants, such as: the queries they entered, the information items they viewed, and their relevance judgments, were logged to allow for quantitative analysis. Participants' confidentiality was maintained throughout the experiment. Before the experiment began, participants were made aware of all feedback sought and were given the option of not participating in the experiment, however, all decided to participate. The participants signed a permission form to ensure that the feedback results could be published. Furthermore, clearance was sought and approved by QUT's ethics committee.

Table 2: The number of relevant elements retrieved by each interface.

	TOPIC NUMBER						
	253	256	257	270	275	284	Average
KO	6	27	64	8	55	45	34.2
NLPX	7	33	22	11	82	5	26.7
Bricks	1	12	22	11	48	5	16.5

Table 3: The ratio of relevant elements retrieved to total number of elements retrieved by each interface.

INTERFACE	TOPIC NUMBER						
	253	256	257	270	275	284	Average
KO	0.084	0.458	0.293	0.097	0.467	0.281	0.280
NLPX	0.069	0.620	0.197	0.162	0.660	0.095	0.300
BRICKS	0.018	0.267	0.215	0.141	0.863	0.192	0.283

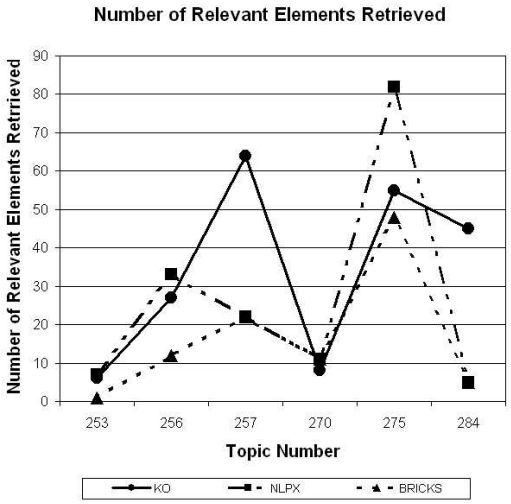

Figure 6: The number of relevant elements retrieved by each interface.

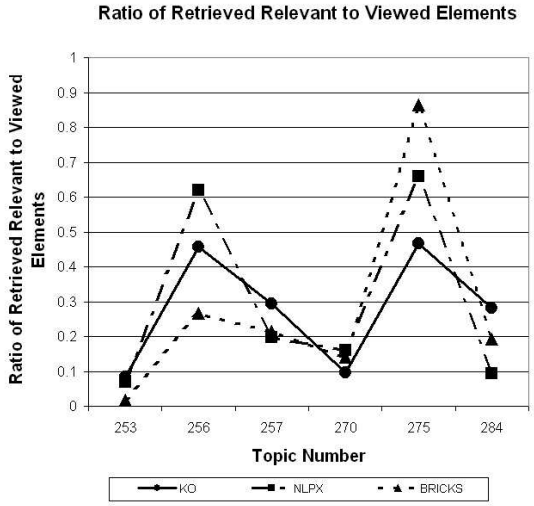

Figure 7: The ratio of relevant elements retrieved to total number of elements retrieved by each interface.

5 Results

5.1 Retrieval Performance

Here, we present the results from our experiments. Our investigation consists of a mixture of quantitative and qualitative analysis. Hence. we present two sets of results, first, the retrieval performance of the three interfaces and second, the results of a survey that examined the usability of both NLPX and Bricks. Official INEX relevance judgements were used in our analysis, thereby, keeping the relevance judgements consistent across participants, since we wanted to narrow the scope of our research to the performance of interfaces.

Table 2 and Figure 6 present the number of relevant elements retrieved by each of the interfaces for the six INEX topics. This is a recall orientated measure. As the results show, there is no significant difference between keywords only and the structured interfaces (NLPX and Bricks), in fact on average the structured interfaces perform worse than keywords only. This finding corresponds to the work of Trotman and Lalmas [8]. Table 3 and Figure 7 present the average ratio of relevant results retrieved to those viewed by users. This is a precision orientated measure. Here, the results for NLPX outperform both the keywords only and Bricks interfaces, however, once again the results are not significant.

5.2 Usability Scores

The second area we investigated was how the two different interfaces, NLPX and Bricks, affected the retrieval experience of the users. Following the experiment we asked the participants five questions about their experience using each interface. For each interface, participants were asked to respond to each question with a rating between 1 and 10 on how well the interface successfully fulfilled a set criterion for that interface. The questions and the average response for all participants presented in Figure 4 and Table 8. The results indicate that the participants did not think that there was much variation between the two interfaces. In particular, the averages for questions two, three and four were almost identical between interfaces. There was a slight difference between interfaces in the averages for questions one and five, however, neither difference is statistically significant.

Table 4: Participants survey results regarding interface usability.

	NLPX	Bricks
Easy to Use	5.313	4.563
Found Relevant Results	4.938	5.000
Ranked Results Highly	4.813	4.750
Accurately expressed my information need	4.500	4.563
Fully expressed my information need	4.00	4.38

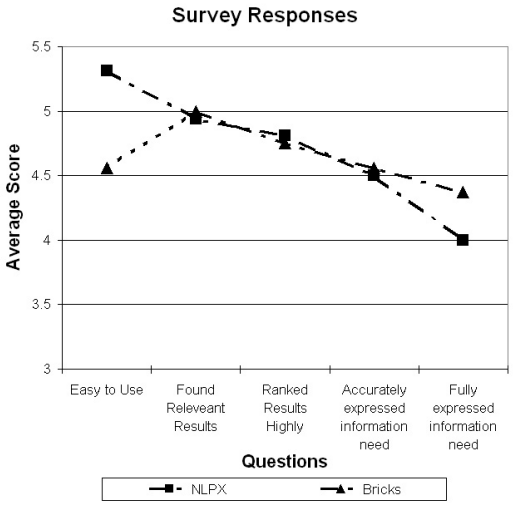

Figure 8: Participants survey results regarding interface usability.

6 Discussion

This research was motivated by a desire to investigate how users interact with XML-IR systems, in particular, how users formulate XML-IR queries. Our experiments focused on the retrieval performance of keyword only and structured query interfaces, and the usability of structured interfaces.

Prior to the study of Trotman and Lalmas [8] it was believed that structured queries would outperform traditional keyword only queries in XML-IR. However, their research disproved this assumption. Our findings correspond to those of Trotman and Lalmas. In fact, there was only one instance were the retrieval performance of NLPX conclusively outperformed that of the keyword only systems. This was for INEX topic 275 presented in Figure 2. Note, that the information need (narrative) for this topic is very direct, specifically asking for sections and abstracts containing certain content items. This may be a reason why it performed so well in our experiment.

We do not know why the keywords only interface performs as strongly as the structured interfaces. It may be, as suggested by Trotman and Lalmas, that users are unable to formulate effective structured queries - that is they are unable to identify which structures in the collection contain relevant information. This possibility is especially pertinent to this experiment since since the participates were uninitiated in the domain of XML retrieval and would therefore be less likely to write effective structured queries than experts. Alternatively it may be as a consequence of INEX IEEE collection since most of the retrieved elements are syntactic (for instance section, paragraph) rather than semantic in nature.

However, we also showed that for structured retrieval, a natural language interface is as effective as a query by template interface. This is important contribution, seeing that this is the first time that a natural language interface has been tested in a usability experiment. According to the user surveys the only difference between the two interfaces was in ease of use and fullness of capturing information need. Users felt that NLPX was easier to use than Bricks, which could show that natural language interfaces are more intuitive than query-by-template interfaces. In contrast, users felt that Bricks captured their information need more fully than NLPX, possibly due to that fact that users were unsure to the degree that NLPX was correctly interpreting their queries.

7 Future Work

Since this was a pilot study on users' interaction with XML-IR system there remains much to be investigated. Here, we outline further research that could be conducted based upon our pilot study.

More participants. Our experiments contained sixteen participants which is not statistically significant for quantitative analysis. However, a larger number of participants (for example fifty to a hundred) would provide a statistically significant number for quantitative analysis while strengthening the qualitative testimony.

Wider pool of participants. The participants in our experiment were post-graduate information technology students which is not representative of the types of users that could possibly use XML-IR systems. In our experiment this was a necessary constraint since the source collection was restricted to IEEE journals. However, if a more general collection was used, then a wider range of participants could be used.

More guidance on how to use NLPX. Participants in our experiment were given minimal guidance on how to use NLPX. This was by design, since we wanted to observe how uninitiated users would

interact with a natural query interface. However, some participants found this disconcerting. It would be interesting to see how further guidance would effect users retrieval experience.

Alternative collection As stated, an alternative source collection would allow for a wider pool of participants. Another justification for an alternative collection is that the IEEE collection was syntactic rather than semantic in nature. Hence, it may not really matter to users if a certain term appears in a paragraph or an abstract. Whereas, if the source collection contained information about movies that it could be very important if the movie was *titled Capote* or if it was *written by Truman Capote*.

Longer time span. Our experiment was conducted within a two hour period. If a similar experiment was conducted over a longer period, for example twelve eighteen months, then more data could be collected and analysed. For instance we could observe if the users interaction with the interfaces changed over time. This would provide valuable information on real users interaction with XML-IR systems.

8 Conclusion

We observed how uninitiated users interact with XML-IR systems and recorded their experience. Our results do not show that incorporating structural hints into queries aids retrieval. However, when structural hints are added to queries our results indicate that users experience with a natural language interface is similar to their experience with a query-bytemplate interface, the current standard for interactive XML-IR systems. These results indicate that further research in is warranted to attain further understanding of XML-IR users needs.

Acknowledgements The authors would like to acknowledge the anonymous participants of the experiments, in particular those participants who were interviewed following the experiment. Without their contribution this work could not been conducted.

References

[1] Norbert Fuhr, Norbert Gövert, Gabriella Kazai and Mounia Lalmas (editors). *INitiative for the Evaluation of XML Retrieval (INEX). Proceedings of the First INEX Workshop. Dagstuhl, Germany, December 8–11, 2002*, ERCIM Workshop Proceedings, Sophia Antipolis, France, March 2003. ERCIM. `http://www.ercim.org/publication/ws-proceedings/INEX2002.pdf`.

[2] Norbert Fuhr, Mounia Lalmas, Saadia Malik and Gabriella Kazai (editors). *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*, Volume 3977 of *Lecture Notes in Computer Science*. Springer, 2006.

[3] Norbert Fuhr, Mounia Lalmas, Saadia Malik and Zoltán Szlávik (editors). *Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Revised Selected Papers*, Volume 3493 of *Lecture Notes in Computer Science*. Springer, 2005.

[4] Shlomo Geva. GPX - Gardens Point XML-IR at INEX 2005. In *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*.

[5] William R. Hersh, Andrew Turpin, Susan Price, Dale Kraemer, Benjamin Chan, Lynetta Sacherek and Daniel Olson. Do batch and user evaluations give the same results? an analysis from the trec-8 interactive track. In *TREC*, 1999.

[6] Richard A. O'Keefe and Andrew Trotman. The simplest query language that could possibly work. In *The Proceedings of the 2003 INEX Workshop*, 2004.

[7] Jovan Pehcevski, James A. Thom, Seyed M. M. Tahaghoghi and Anne-Marie Vercoustre. Hybrid xml retrieval revisited. In Fuhr et al. [2], pages 153–167.

[8] Andrew Trotman and Mounia Lalmas. Why structural hints in queries do not help xml-retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 711–712, New York, NY, USA, 2006. ACM Press.

[9] R. Van Zwol, J. Baas, H. Van Oostendorp and F. Wiering. Query formulation for xml retrieval with bricks. In *Proceedings of the INEX 2005 Workshop on Element Retrieval Methodology*, 2005.

[10] Ellen M. Voorhess and Donna K. Harman. *TREC: Experiment and Evaluation in Information Retrieval*. MIT Press, Cambridge, Massachusetts, 2005.

[11] Alan Woodley and Shlomo Geva. Nlpx at inex 2004. In *Advances in XML Information Retrieval, Third International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2004, Dagstuhl Castle, Germany, December 6-8, 2004, Revised Selected Papers*, pages 382–394, 2005.

[12] Alan Woodley and Shlomo Geva. Nlpx at inex 2005. In *Advances in XML Information Retrieval and Evaluation, 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005, Revised Selected Papers*, pages 358–372, 2006.