
© Copyright 2008 Springer
GPX@INEX2007: Ad-hoc Queries and Automated Link Discovery in the Wikipedia

Shlomo Geva
Faculty of IT
Queensland University of Technology
Brisbane, Australia
s.geva@qut.edu.au

Abstract The INEX 2007 evaluation was based on the Wikipedia collection in XML format. In this paper we describe some modifications to the GPX search engine and the approach taken in the Ad-hoc and the Link-the-Wiki tracks. The GPX retrieval strategy is based on the construction of a collection sub-tree, consisting of all nodes that contain one or more of the search terms. Nodes containing search terms are assigned a score using the GPX ranking scheme which incorporates an extended TF-IDF variant. In earlier version of GPX scores were recursively propagated from text containing nodes, through ancestors, all the way to the document root of the XML tree. In this paper we describe a simplification whereby the score of each node is computed directly, doing away with the score propagation mechanism. Preliminary results indicate improved performance. The GPX search engine was used in the Link-the-Wiki track to identify prospective incoming links to new Wikipedia pages. We also describe a simple and efficient approach to the identification of prospective outgoing links in new Wikipedia pages. We present preliminary evaluation results.

1. The GPX Search Engine

For the sake of completeness we provide a very brief description of GPX. The reader is referred to earlier papers on GPX in INEX previous proceedings for a more complete description. The search engine is based on XPath inverted lists. For each term in the collection we maintain an inverted list of XPath specifications. This includes the file name, the absolute XPath identifying a specific XML element, and the term position within the element. The actual data structure is designed for efficient storage and retrieval of the inverted list which are considerably less concise by comparison with basic text retrieval inverted lists. We briefly describe the data structure, then we describe the node scoring calculation, and finally we present the results.

2. GPX Inverted List Representation

The GPX search engine is using a relational database implementation (Apache Derby) to implement an inverted list data structure. It is a compromise solution provides the convenience of a DBMS at the cost of somewhat reduced performance which may otherwise be possible.

Consider the XPath:

\[
/\text{article}[1]/\text{bdy}[1]/\text{sec}[5]/\text{p}[3]
\]

This could be represented by two expressions, a Tag-set and an Index-set:

Tag-set: \text{article/bdy/sec/p}

Index-Set: 1/1/5/3

The original XPath can be reconstructed from the tag-set and the index-set. It turns out that there are over 48,000 unique tag-sets, and about 500,000 unique index-sets in the collection. We assign to each tag set and each index-set a hash code and create auxiliary database tables mapping the hash-codes to the corresponding tag-set and index-set entries. These hash tables are small enough to be held in memory and so decoding is efficient.
The GPX database tables are then:

- **Term-Context** = \{ Term-ID, File-ID, XPath-Tag-ID, XPath-IDX-ID, Position \}
- **Terms** = \{ Term, Term-ID \}
- **Files** = \{ File-Name, File-ID \}
- **TagSet** = \{ XPath-Tag-ID, Tag-Set \}
- **IndexSet** = \{ XPath-IDX-ID, Index-Set \}
- **XPathSize** = \{ XPath-ID, Node-Size \}

Given a search term the database can be efficiently accessed to obtain an inverted list containing the context of all instances where the term is used (identified by File Name, full XPath, and term position). Having retrieved a set of inverted lists, one for each term in the query, the lists are merged so as to keep count of query terms in each node and also keeping the term positions. Stop words are actually indexed, but too frequent terms are ignored by applying a run-time stop-word frequency threshold of 300,000. We also used plural/singular expansion of query terms. We have found that - on average – the use of a Porter stemmer is not adding to system performance and so it was not used.

Having collected all the nodes that contain at least one query term the system proceeds to compute node scores. Calculation of node relevance score from its content is based on a variation of TF-IDF. We used the inverse collection frequency of terms rather than the inverse document frequency (TF-ICF). The score is then moderated by a step function of the number of unique terms contained within the node. The more unique terms the higher the score. The score is further moderated by the proximity within which the terms are found. Additionally, the scores of all article nodes that contained query terms in the <name> node were further increased. All this can be calculated with the information in the inverted lists.

### 3. Calculation of Text Nodes Score

GPX 2007 deviates significantly from earlier with respect to the way that ancestor node scores are calculated. For clarity we shall refer to GPX-2007 to denote the current system and GPX to denote the older system. In the earlier version GPX computed node scores on the basis of direct text content (having a text node in the DOM model) and then the scores were propagated upwards in the XML tree. GPX accumulated all children node scores for a parent and reduced the score by a decay factor (typically about 0.7) to account for reduced specificity as one moved upwards in the XML tree. In GPX 2007 the scores are computed directly from the node text content, direct, or indirect. That means that any node is scored by the text it contains regardless of whether it has a direct text node in the DOM representation – all the text in the node and its descendents is used.

Naturally, nodes closer to the root could receive a higher score on account of more query terms in descendent nodes. A common variation to TF-IDF is to normalise the score by taking into account the document size. The motivation there is to account for the increased probability of finding query terms in larger documents and hence biasing the selection towards larger documents. The motivation here is similar with a slight twist. Node normalisation in the XML score calculation is motivated by the need to compensate for the reduced specificity of larger nodes. We are aiming for focused retrieval and look for nodes of “just the right size” (whatever that may be.) Node normalisation introduces a penalty in a parent node that contains large amounts of irrelevant text in descendent nodes and which do not contribute towards an increased score. However, when two nodes have a similar size but contain different amount of relevant text then the more relevant node will score higher.

But there is another twist here. We also know that nodes that are too small are unlikely to satisfy a user information need (except perhaps in factoid type QA). At least with the Wikipedia we know that the most common element selected by assessors is a paragraph (or passage). Very small passages are not common in the qrels of past experiments. Therefore, we do not want to normalise the scores of too small nodes thereby unduly increasing their score relative to otherwise similarly scoring nodes which are somewhat larger. Node scores are normalised by dividing the raw score by the node size (measured as the number terms), but all nodes with size of below 75 terms are normalised by 75. This heuristic is convenient in the XML case because when breaking ties in node selection (focused
retrieval) we prefer the ancestor to the descendant when the scores are equal. This means that we prefer parent nodes as long as the parent is larger than the descendant and below 75 terms in size. For example, this means that a very deep XML branch with no breadth will be collapsed to an ancestor of up to size 75 terms (if such exists). So in summary, node size normalisation is biasing the selection towards passages of 75 terms, both from above and from below. We experimented with other values for node size from 50 to 150 with little difference in results. More careful sensitivity analysis is still pending.

Since GPX 2007 we now computes node scores over much larger text segments it is necessary to take account of term proximity. The intuition is that we should award higher scores to nodes in which search terms are found in closer proximity to each other. In earlier versions of GPX this was not critical since node scores were computed at text nodes and these were typically paragraphs, titles, captions, and other such relatively small nodes. A proximity function was defined and incorporated into the score calculation. So finally we have the following score calculation:

**Equation 1**: Calculation of $S$, node size for normalisation

$$S = \begin{cases} \text{NodeSize} > 75 \\ 75 \end{cases} \leq 75$$

The value of $S$, the node size for the purpose of normalization, is thus equal to 75 for nodes smaller than 75 terms, but taken as the actual node size for nodes with more terms.

**Equation 2**: Calculation of $P$, node terms proximity score

$$Pr = 10\sum_{i=1}^{n} \exp\left(-\frac{P_i - P_{i+1} + 1}{5}\right)^2$$

Here terms are processed in the order in which they appear in the text node. $P_i$ is the position of term $i$ in the text node. This is a Gaussian function with a maximum value of 10 and decaying exponentially with increased term distance between successive terms. The function is depicted in Figure 1. Note that in practice, a table lookup is more efficient than the numerical calculation.

![Proximity score as a function of term separation](image_url)
Equation 3: Calculation of element relevance score from its content

\[ L = \frac{\Pr}{S} K^{n-1} \sum_{i=1}^{n} \frac{t_i}{f_i} \]  

Here \( n \) is the count of unique query terms contained within the element, and \( K \) is a small integer (we used \( K=5 \)). The term \( K^{n-1} \) is a step function which scales up the score of elements having multiple distinct query terms. This heuristic of rewarding the appearance of multiple distinct terms can conversely be viewed as taking more strongly into account the absence of query terms in a document. Here it is done by rewarding elements that do contain more distinct query terms. The system is not sensitive to the value of \( K \) and a value of \( k=5 \) is adequate. The summation is performed over all \( n \) terms that are found within the element where \( t_i \) is the frequency of the \( i^{th} \) query term in the element and \( f_i \) is the frequency of the \( i^{th} \) query term in the collection.

Finally, nodes that contain query terms that are preceded by a minus sign (undesirable) are eliminated.

At this point we have computed the score of all (overlapping) nodes in each article that contains query terms. The score of the <article> node itself is then added to all nodes in the article. This lifts the scores of all nodes that appear in a high scoring article. The intuition is that an article with many scoring nodes is more likely to be relevant and so all its scoring elements are ranked higher on account of more scoring nodes appearing in the same article. Without this modification, two similar nodes, one being an isolated instance of a relevant node in an article, and the other being one of many relevant nodes in an article, would receive a similar score.

Although more analysis of the results is required, preliminary results suggest an improved performance in GPX. The runs labelled RIC_04 and BIC_04 were produced with the 2006 GPX version (score propagation) while BIC_07 and RIC_07 were run with the GPX_07 version with direct score calculation. The GPX 07 version seems to perform better than the earlier GPX version over almost all reported measures. It does not require any magic numbers (decay constants) and is therefore more appealing.

4. Ad-Hoc retrieval tasks

The Ad-Hoc track at INEX 2007 consisted of 3 tasks – Focused, Relevant in Context, and Best in Context. These tasks are described elsewhere in this proceedings collection. We briefly describe the approach taken to each of the tasks in our best performing run.

4.1 Focused Retrieval

Focused Retrieval starts with the thorough results recall base. Within each article the highest scoring elements on a path are selected by keeping only elements that have a higher score than any of their descendents or ancestors. The submission consists of the remaining overlap free focused elements, sorted by descending score.

4.2 Relevant in Context (RIC)

The objective of the task was to balance article retrieval and element retrieval. Whole articles are first ranked in descending order of relevance and within each article a set of non-overlapping most focused elements are grouped. We have used the focused results, which were overlap free already, but grouped the elements within articles and sorted the articles by article score.

4.3 Best in Context (BIC)

We tested a trivial approach here – we simply kept the highest scoring element in each document appearing in the focused recall base.
5. Link the Wiki

The Link the Wiki task is described in detail elsewhere in this proceedings collection. The objective of this task was to identify a set of incoming links and a set of outgoing links for new Wikipedia pages. In practice, the topics were existing Wikipedia pages that were stripped of exiting links. The links were only at the article-to-article level. We adopted rather simple approaches.

5.1 Incoming links

Incoming links were identified by using the GPX search engine to search for elements that were about the topic name element. For each topic the name element was used to construct a standard NEXI query:

```
//article[about(.,'name')]
```

We have used the SCAS task setting whereby the results were interpreted strictly. In this case it only means that articles nodes were returned. This was sufficient since only article-to-article links were needed. Results were ordered by article score with the more likely relevant articles returned earlier in the list. The process took an average of 8.5 seconds per topic.

5.2 Outgoing links

We have adopted a very simple approach to this task. All existing page names in the Wikipedia were loaded into an in-memory hash table (with collision resolution). With 660,000 articles this is not an onerous task. The identification of potential links was based on a systematic search for anchor text that matches existing page names. In the first stage we have extracted the text of the topic (eliminating all markup information.) Prospective anchors for outgoing links were identified by running a window over the topic text and looking for matching page names in the collection. The window size varied from 8 words down to 1 word, and included stop words. Longer anchors were ranked higher than shorter ones, motivated by the trivial observation that the system was less likely to hit on a longer page name by accident. A naïve approach perhaps, but quite effective as it turns out. The process is purely computational and does not incur any I/O operations. The process took an average of 0.6 seconds per topic.
5. Results

The GPX system performed well and produced particularly good results in the Relevant in Context and Best in Context tasks of the Ad-hoc track, and in the Link-the-Wiki track.

5.1 Ad Hoc retrieval

<table>
<thead>
<tr>
<th>Task</th>
<th>Best MApP</th>
<th>Best GPX MApP</th>
<th>Run rank</th>
<th>System rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>Relevant in Context</td>
<td>0.1013</td>
<td>0.0975</td>
<td>6/66</td>
<td>2/17</td>
</tr>
<tr>
<td>Best in Context</td>
<td>0.1951</td>
<td>0.1823</td>
<td>4/71</td>
<td>3/19</td>
</tr>
<tr>
<td>Focused</td>
<td>0.4259</td>
<td>0.3842</td>
<td>13/79</td>
<td>7/25</td>
</tr>
</tbody>
</table>

Relatively good results were achieved in terms of precision at early recall levels on most of the tasks. Complete results sets are available on the INEX web site: http://inex.is.informatik.uni-duisburg.de/2007/adhoc-protected/Evaluation.html

5.2 Link-theWiki
In the Link-the-Wiki task apparently good results were achieved but the significance of this had not yet been established given the nature of the evaluation (no manual assessment was involved). There is also no baseline for comparison since this is the first time that the task was run.

Complete results sets are available on the INEX web site:
http://inex.is.informatik.uni-duisburg.de/2007/lw-protected/results.html
and also in the paper describing the Link the Wiki task in these proceedings.

6. References
