Enhancing Keyword Search with a Keyphrase Index

Miro Lehtonen[1] and Antoine Doucet[1,2]

[1] Department of Computer Science
P. O. Box 68 (Gustaf Hällströmin katu 2b)
FI–00014 University of Helsinki
Finland
{Miro.Lehtonen,Antoine.Doucet} @cs.helsinki.fi
[2] GREYC CNRS UMR 6072,
University of Caen Lower Normandy
F-14032 Caen Cedex
France
Antoine.Doucet @info.unicaen.fr

Abstract. Combining evidence of relevance coming from two sources — a keyword index and a keyphrase index — has been a fundamental part of our INEX-related experiments on XML Retrieval over the past years. In 2008, we focused on improving the quality of the keyphrase index and finding better ways to use it together with the keyword index even when processing non-phrase queries. We also updated our implementation of the word index which now uses a state-of-the-art scoring function for estimating the relevance of XML elements. Compared to the results from previous years, the improvements turned out to be successful in the INEX 2008 ad hoc track evaluation of the focused retrieval task.

1 Introduction

The interest in developing methods for keyphrase search has decreased recently in the INEX community partly because most of the queries are not keyphrase queries [1]. However, we believe that indexing interesting phrases found in the XML documents can be useful even when processing non-phrase queries. As the XML version of the Wikipedia is full of marked-up phrases, we have been motivated to work on the quality of the phrase index, as well, in order to capture those word sequences that document authors really intended to be phrases.

In the previous years, our ad hoc track results have not been at the same level with the best ad hoc track results. We believed that the reason lay in the keyword index and the tfidf scoring function because the top results were achieved with the probabilistic retrieval model. Lesson learned: we introduced BM25 as the new scoring function for the keyword index. The latest results of the INEX 2008 evaluation show great improvement from previous years. How much the improvement is due to the state-of-the-art scoring function and how much to the improved phrase index is still unclear, though.

This article is organised as follows. Section 2 describes our IR system as it was implemented in 2008. In Section 3, we show how the keyphrases are extracted from the document collection into a keyphrase index. Section 4 details the scoring methods for both the word index and the keyphrase index. The results of the evaluation are presented in Section 5, and finally, we draw conclusions and directions for future work in Section 6.

2 System description

Our system was built on the EXTIRP architecture [2]. With one pass of the whole collection XML documents, we select a set of disjoint XML fragments which are indexed as an atomic unit of text. We can apply the paradigm of document retrieval to the fragments because they do not overlap. The indexed fragments of XML are entire XML elements in a predefined size range of 150–7,000 XML characters. The total number of indexed fragments is 1,022,062 which is just over 1.5 fragments per article. The number is relatively low because most of the Wikipedia articles are too small to be further divided into smaller fragments.

Two separate inverted indices are built for the fragments. A *word index* is created after punctuation and stopwords are removed, case folded, and the remaining words are stemmed with the Porter algorithm [3]. The *phrase index* where a phrase is defined as a Maximal Frequent Sequence (MFS) [4] is described in Section 3.2.

3 The anatomy of a keyphrase index

Building a keyphrase index starts from finding or detecting the word sequences that should be considered keyphrases. As we are indexing hypertextual XML documents, it is natural to use the characteristics of hypertext documents and the markup language in the analysis as we detect passages that are potentially indexed keyphrases. The analysis is followed by a text mining method for extracting the Maximal Frequent Sequences from the word sequence.

3.1 Phrase detection and replication

Most of the XML markup in the Wikipedia articles describes either the presentation of the content or the hyperlink structure of the corpus, both of which show as mixed content with inline level XML elements. In these cases, the start and end tags of the inline level elements denote the start and the end of a word sequence that we call an *inline phrase*. These phrases include the anchor texts of hyperlinks as well as phrases with added emphasis, e.g., italicized passages. An exact definition for the XML structures that qualify was presented at the INEX 2007 workshop [5]. Intuitively, the inline phrases are highly similar to the multiword sequences that text mining algorithms extract from plain text documents. Therefore, the tags of the inline elements are strong markers of potential phrase

boundaries. Because phrase extraction algorithms operate on word sequences without XML, we incorporate the explicit phrase marking tags into the word sequence by replicating the qualified inline phrases.

Considering the effect of replication, we only look at the character data (CDATA) as the tags and other XML markup are parsed and removed before phrase extraction. The most obvious effect is the increase in phrase frequency of the replicated inline phrases with a similar side effect on the individual words they compose of. Moreover, the distance between the words preceding and following the phrase increases, which makes the phrase boundaries more explicit to those phrase extraction algorithms that allow gaps in the multiword sequences.

Duplicating the inline phrases lead to a 10–15% improvement in the MAiP on the INEX 2007 topics [6], but more recent experiments where the phrases were replicated three times have shown even further improvement when tested on the same topics. Note that these results depend on the phrase extraction algorithm and that other algorithms than ours may lead to different figures. Anyway, we chose to see if the triplication of the inline phrases works on the INEX 2008 topics, as well, and built the phrase index correspondingly.

3.2 MFS extraction

The *phrase index* is based on Maximal Frequent Sequences (MFS) [4]. A sequence is said to be frequent if it occurs more often than a given sentence frequency threshold. It is said to be maximal if no other word can be inserted into the sequence without reducing the frequency below the threshold. This permits to obtain a compact set of document descriptors, that we use to build a phrase index of the collection.

The frequency threshold is decided experimentally, because of the computational complexity of the algorithm. Although lower values for the threshold produce more MFSs, the computation itself would take too long to be practical.

To be able to extract more descriptors, we clustered the XML fragments of the Wikipedia collection into 250 disjoint clusters. This permits to fasten the extraction process and to locally lower frequency threshold values. The result is a phrasal description of the document collection that is enhanced both in terms of quality and quantity of the descriptors. The drawback of this approach is a less compact document description. To perform this divide-and-conquer extraction of MFS, we used the $MFS\_MineSweep$ algorithm which is discussed in full detail in [7].

3.3 Arguments for two phrase extraction methods

Extracting the marked-up phrases and computing the frequent word sequences are both adequate methods for finding interesting phrases in hypertext documents. However, as our method that utilizes XML markup ignores the tag names, it generates a substantial amount of noise even though most marked phrases are captured. Examples of this noise include infrequent phrases that occur in few documents and various passages where the typeface differs from the

surrounding content. Moreover, if words are inserted in the middle of a phrase, it shows as multiple marked up words instead of a single phrase with gaps in it. Therefore, the markup is not as reliable indicator of a phrase as the statistical occurrences of word sequences.

Maximal Frequent Sequences is a rather stable definition for an indexed phrase. First, the extracted word sequences are statistically frequent. Second, natural variation in the sequences is allowed in the form of gaps within the phrase. Replication of the marked up phrases changes the word sequence where the maximal sequences are computed. Incorporating the markup-based component in the text mining algorithm further stabilizes the method, which on the whole improves the quality of the phrase index [6].

4 Scoring XML fragments

When processing the queries, we compute two separate RSV values that are later combined: a `Word_RSV` value based on a word index, and an `MFS_RSV` value based on the phrase index.

The `Word_RSV` is calculated using Okapi BM25 as implemented in the Lemur Toolkit [8], while the `MFS_RSV` is computed through loose phrase matching, in an identical way as in earlier versions of our system [9]. An exact match of the query phrase is not required, but gaps between query words and a different order of the query terms do contribute less to the score than an exact match.

The combination of both RSV values is done as follows. First, both values are normalized into the [0,1] range, using *Max Norm*, as presented by Lee [10]. Following this step, both RSVs are aggregated into a single RSV through linear interpolation, so that the aggregated RSV = α * `Word_RSV` + β * `MFS_RSV`.

In previous INEX participations, α was the number of distinct query terms and β was the number of distinct query terms in the query phrases. Post INEX 2007 experiments showed better performance with absolute values throughout the topic set, and we have decided to rely on such a new setting for our 2008 experiments as well, with α ranging between 92 and 94 and β = 100 - α.

The relatively low value of β is due to the fact that the phrase index only contains words that are frequent enough in phrasal context, that is, frequent enough in conjunction with at least one other word. Important words that do not co-occur sequentially do not appear in the phrasal index. For this reason, the phrasal RSV not self-sufficient and should be perceived as a complement of the word RSV.

5 Results

We submitted three runs for the ad hoc track task of focused retrieval. The configurations of the submitted runs were based on experiments on the ad hoc track topics of INEX 2007, according to which the best proportion of weight given to terms and phrases would be around 92:8–94:6. The weight is given to

the word index component is part of the Run ID. The initial results including 70 topics with assessments are shown in Table 1.

Table 1. Evaluation of our three official runs submitted for the focused retrieval task.

Run ID	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP
UHel-Run1-92	0.6920	0.6534	0.5568	0.4996	0.2256
UHel-Run2-93	0.7030	0.6645	0.5583	0.5028	0.2271
UHel-Run3-94	0.7109	0.6648	0.5558	0.5044	0.2268

None of the submitted runs is significantly better than the other two runs although the interpolated precision does show moderately different figures at the lowest levels of recall. However, the results are similar to those of our earlier experiments on INEX 2007 topics where the precision peaks when α is set between 92 and 94. Compared to the peak values, if α is set to 0, precision drops by over 30%, whereas setting α to 100 (BM25 baseline) results in a modest decline of 1–5% depending on the recall point. However, we have not yet conducted this experiment on the 2008 topics and can thus not confirm the previous observations.

6 Conclusion and future work

The biggest change in our system from 2007 took place in the scoring function that contributes over 90% of the total relevance score of each XML fragment. We discarded tfidf and replaced it with BM25 which assumes the probabilistic model for information retrieval. Thanks to that update, our results are now comparable with the best results overall. The results also confirm that our phrase index slightly improves precision from a baseline where BM25 is the only scoring function as the optimal weight given to the phrase score is around 7%. Investigating whether the weights should be different for different types of queries is part of our future work.

References

1. Doucet, A., Lehtonen, M.: Let's phrase it: INEX topics need keyphrases. In: Proceedings of the SIGIR 2008 Workshop on Focused Retrieval. (2008) 9–14
2. Lehtonen, M., Doucet, A.: Extirp: Baseline retrieval from wikipedia. In Malik, S., Trotman, A., Lalmas, M., Fuhr, N., eds.: Comparative Evaluation of XML Information Retrieval Systems. Volume 4518 of Lecture Notes in Computer Science., Springer (2007) 119–124
3. Porter, M.F.: An algorithm for suffix stripping. Program **14** (1980) 130–137
4. Ahonen-Myka, H.: Finding all frequent maximal sequences in text. In Mladenic, D., Grobelnik, M., eds.: Proceedings of the 16th International Conference on Machine Learning ICML-99 Workshop on Machine Learning in Text Data Analysis, Ljubljana, Slovenia, J. Stefan Institute (1999) 11–17

5. Lehtonen, M., Doucet, A.: Phrase detection in the Wikipedia. In: Focused access to XML documents, 6th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2007. Volume 4862/2008 of Lecture Notes in Computer Science. (2008) 114–121
6. Lehtonen, M., Doucet, A.: XML-aided phrase indexing for hypertext documents. In: SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2008) 843–844
7. Doucet, A., Ahonen-Myka, H.: Fast extraction of discontiguous sequences in text: a new approach based on maximal frequent sequences. In: Proceedings of IS-LTC 2006, "Information Society, Language Technology Conference". (2006) 186–191
8. Lemur: Lemur toolkit for language modeling and ir (2003)
9. Doucet, A., Aunimo, L., Lehtonen, M., Petit, R.: Accurate Retrieval of XML Document Fragments using EXTIRP. In: INEX 2003 Workshop Proceedings, Schloss Dagstuhl, Germany (2003) 73–80
10. Lee, J.H.: Combining multiple evidence from different properties of weighting schemes. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press (1995) 180–188