Information Retrieval System based on Phrase Indices

Hatem Nassrat
Dalhousie University - Faculty of Computer Science

December 8, 2008

Abstract

Information retrieval systems deal with retrieving relevant document from document sets with high precision. The quest in finding the perfect (or near perfect) system is arguably a never ending one. The current information retrieval tools has provided us with tools that have taken us literally half way to our goal. We may argue that we have reached our limits with the current designs, and need to find ways to complement our system to reach perfection. For that reason we propose a study, were we augment the current popular indexing scheme used in most IR systems with phrase indices in order to achieve a better information retrieval system.

1 Introduction

The science of information retrieval (IR) deals with retrieving relevant documents from various corpora. The general strategy in IR systems is to create an inverted file of terms mapped to documents in which they appear in. There exists various strategies to query such inverted indices. Vector Space strategy is a popular information retrieval technique used to retrieve relevant documents. This technique is quite robust and returns documents with a satisfactory level of precision.

This study analyses the effect of extending the standard indices used in standard vector space IR systems to use Multi Keywords or Phrases. However, the term phrases here is used liberally to mean N-grams of words, rather than linguistically valid phrases. The effect of this extension in terms of precision and recall is measured and presented in the following sections of the paper.

2 Related Work

After a brief investigation of the literature it has been found that there has been various attempts to create phrase indices to improve precision and recall. [14][4][12][5][11] have utilized next word indices in building their phrase indices. Generally each of these studies

crude china	coffee colombia	interest uk
earn italy	grain china	acq uk
cocoa usa	yen japan	corn usa
rapeseed japan	carcass usa	interest usa
oilseed china	wheat ussr	trade japan

Figure 1: Queries using the Topic fields in Reuters 21578

have used some linguistic method to choose whether or not to include the phrases in their index.

Since users of many IR systems tend to use short queries [10], queries may not form linguistically valid queries. Therefore we have decided to undertake this study which does not discriminate linguistically between phrases while, and moreover, places all the terms in a single index for ease of querying. The research produced in [14] have briefly mentioned that such an indexing scheme would be costly in terms of disk space. Since disk space is getting cheaper and cheaper every day, this reason was not a deal breaker for this research quest. The thesis [9] have adopted a similar method, yet they have tightly coupled their system with the peer to peer framework and therefore only deal with small pieces of information, namely file names.

3 Method

The aim of this study is to evaluate the benefit of using phrases in order to increase the performance, in terms of precision and recall, of an IR system. To do so, a base IR system was implemented to serve as a standard Vector Space search engine. This system was further augmented to utilize the proposed phrase index and the difference in performance was measured.

3.1 Dataset

For this study the Reuters 21578 dataset is utilized[2]. All the text under the TEXT tags are indexed for each document. Words in document titles are given a higher weight. Using the TOPICS and PLACES tags fifteen two word queries were generated along with their relevant document sets. Documents were labelled relevant if they contain within the TOPICS and PLACES tags any of the query words. These tags are not under the TEXT tag and therefore were not used for indexing as to not skew the results.

Figure 1 displays the queries that were chosen from the dataset for evaluation purposes. These queries were divided into three groups. The right most contains a small relevant dataset (average of 10 relevant documents), the left most containing a large number of relevant documents (an average of 140 documents) while the middle list contains a moderate number of relevant documents (an average of 40 documents).

3.2 Inverted Index

The base system utilized a basic inverted index. This index mapped single terms to a mapping between documents to the frequency of the term in the document. The dataset being utilized (Section 3.1), contains a single boundary between the *Title* and the article *Text*. The text within the article *Title* has been given more emphasis that the standard *Text*, to do so each word in the *Title* has been given the weight of 10 standard words. Moreover, the standard procedures of stemming [13] and stop word removal [1] have been performed on the dataset.

This inverted index has been built utilizing a B-Tree, the use of the B-Tree based data structure over a simple hash data structure will be apparent later in this section. We have used the Tokyo Cabinet (TC) DBM implementation for the indices of both systems (base and augmented system). The reason for this choice is of the shear performance advantage that *TC* has over the other DBM clones, namely *Berkeley DB* and *GDBM* in both the Hash and B-Tree structures [7].

3.2.1 Augmentation

The standard index has been augmented by adding phrases to this index. This has been done by processing words of a documents which appear in a given proximity as being members of a phrase. This has been done through the utilization of a window based approach.

A window denotes the bounds within which the first word in the window can join with potential partners to form a phrase. The window of a given length N, could span multiple document boundaries, such as joining words from the *Title* with normal words from the *Text* in documents of the dataset (Section 3.1). The first word of the document is selected along with its combination to the one, two, three, up to M combinations of the remaining words in the window. Words that make up a phrase are sorted lexicographically. Each of these potential phrases are stored in an in memory data structure, along with the index of the rightmost word used in the phrase. Equation 1 displays the count of the phrases from a single instance of described window.

$$1 + \sum_{i=1}^{M} \binom{N-1}{i} \tag{1}$$

$$(1 + \sum_{i=1}^{M} \binom{N-1}{i}) \times (X + Y) \tag{2}$$

Phrases are flushed from memory only when the rightmost word in the phrase has created its own phrases and left the window. While a phrase is in memory we only keep a single copy of a phrase, if a duplicate phrase is to be inserted the better of the two phrases is chosen. The better phrase is chosen by calculating the proximity between the words of a phrase, and choosing the instance that is of the smallest cumulative proximity. Similarly words cannot be combined to its identical term, i.e. only phrases that contain unique terms are kept. These simple steps are used to penalize documents with many repeated terms as

they will not have many phrases generated. More importantly this reduces the number of useless terms that are gathered by this indexing scheme. Equation 2 provides an upper limit to the number of words generated from a document that has X words in the *Title* and Y words in the *Text*.

Initially all the phrases were generated in one shot. However, the indices became very large quickly. To overcome this issue the apriori methodology has been applied to the phrase index generation. First all the single word phrases are generated. In the following step phrases of two words are accepted if and only if a word within this phrase appeared more than K times in the previous step. This is done for every step up to the M max phrase length. The K denotes the minimum support count of the apriori algorithm.

All the phrases are stored as keys within a single inverted index. Combined with a B-Tree data structure, this phrase index can be utilized to gather query expansions or suggestions for the user of such an IR system.

3.3 Retrieval

The retrieval mechanism for the base system is identical to the one used in many vector space based IR systems. In this study we utilize the normalized TFIDF term weighting scheme [3], to generate the document vectors. As for the query, the terms are given a simple weight of one, as to reduce the effects of a different scheme as to measure the effect that is being studied.

Querying the phrase (multi-keyword) index, is similar to querying the single word index. The query is processed in the same way the document was indexed, following each generated phrase is given a term weight of one.

4 Results

To evaluate the advantage of using phrase indices, the precision at 1, 10, and 50 has been calculated. Moreover, the mean average precision (MAP) and recall has been calculated 1. Other factors such as index sizes and retrieval speeds between the two search engines are discussed in order to verify feasibility of such an index.

For the phrase based index we have used a window size of 10 and a max phrase size of 4 and an apriori support count of 50. The following subsections display the comparison between the base system and the phrase based system.

4.1 Size

Using the given dataset the single word index contains 38,067 unique stemmed words. The phrase index contains 15,178,734 entries (phrases of stemmed words). These are made of 38,067 single words, 2,615,008 word pairs, 8,726,517 word triples, and 3,799,142 phrases of four word length.

On a Unix EXT 3 file system, the Tokyo Cabinet inverted database for the base system has a size of 9.7 MB (as reported by the *du* utility). On the other hand, the phrase based index built using the TC *BZ2* encoding flag has a size of 281 MB.

The number of entries as expected varies greatly between the two systems. The phrase based system, even after apriori reduction, is 400 times larger than the single word phrase system. The file size of the final phrase index is approximately 30 times the size of the base system's index.

4.2 Speed

There is a speed difference between the two systems while searching either systems depending on the number of words in the query. The speed curves for each of the systems can be seen in Figure 2 Since queries are often a couple of words in length, yet they may sometimes be longer [10], the speed variation may not be significant. Moreover, as can be seen from the figure the deviation starts occuring for queries of length greater than four and this aspect may be ignored. This speed variation can be expected due to the query methodology in the proposed system where all phrases are generated from each query.

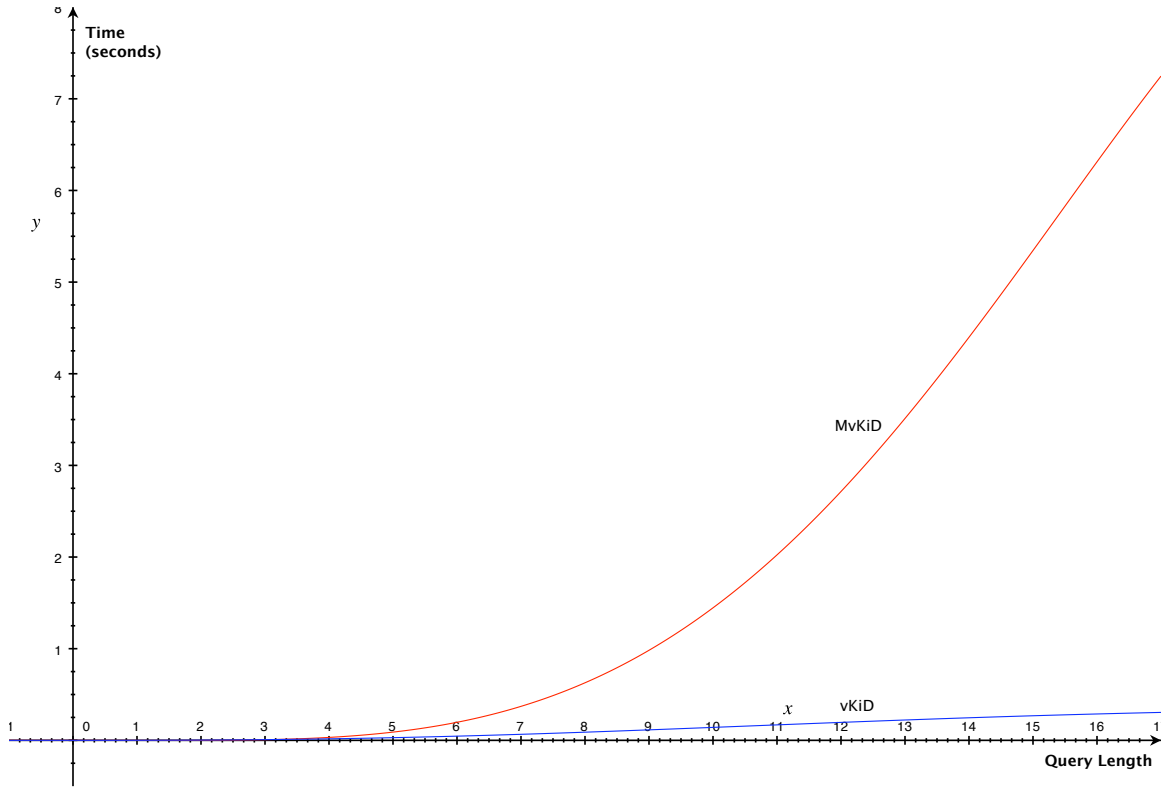

Figure 2: Speed comparison varying query lengths

Small pre-tagged result set					
Query	P@1	P@10	P@50	MAP	Recall
china crude	0.0%	20.0%	4.0%	8.7%	100.0%
china oilseed	0.0%	10.0%	4.0%	7.0%	100.0%
cocoa usa	0.0%	0.0%	4.0%	4.4%	90.0%
earn italy	0.0%	10.0%	2.0%	2.6%	62.5%
japan rapeseed	100.0%	60.0%	12.0%	41.6%	100.0%
Medium pre-tagged result set					
Query	P@1	P@10	P@50	MAP	Recall
carcass usa	0.0%	0.0%	0.0%	0.1%	4.5%
china grain	100.0%	90.0%	36.0%	40.5%	100.0%
coffee colombia	100.0%	100.0%	62.0%	91.0%	100.0%
japan yen	0.0%	20.0%	6.0%	9.7%	100.0%
ussr wheat	100.0%	70.0%	50.0%	51.0%	100.0%
Large pre-tagged result set					
Query	P@1	P@10	P@50	MAP	Recall
acq uk	0.0%	0.0%	8.0%	1.7%	28.9%
corn usa	0.0%	0.0%	2.0%	31.4%	94.2%
interest uk	0.0%	20.0%	18.0%	14.3%	94.1%
interest usa	0.0%	0.0%	0.0%	0.8%	37.7%
japan trade	100.0%	50.0%	26.0%	29.5%	98.8%

Table 1: Precision and Recall for the base system

4.3 Precesion and Recall

Phrase index augmentation have increased the precision of the base system. Table 1 displays the gathered precision and recall values for the base system, and table 2 displays the results after augmentation.

As can be seen from the tables there is an increase in the precision by augmenting the system with some queries while no difference in the recall. This result was expected as there should be no reason for the recall to increase or decrease except for the threshold used in the cosine similarity measure in the vector space search engine. This measure has been tuned to accomodate due to the larger number of terms generated for a query in the second system compared to a query to the base system.

The actual increases between both systems varied according to the query being annalyzed. When looking at the query groups (Figure 1). The main advantage was visible with queries that had a larger number of relevant results, where the average increase can be seen in table 3. As for the other two groups, there is some fluctuation between positive and negative, as can be seen in table 3, have a larger positive impact than a negative impact in general. Nevertheless, the difference in the smaller sets is negligable and cannot be accounted as a plus for the phrase index system.

Small pre-tagged result set					
Query	P@1	P@10	P@50	MAP	Recall
china crude	0.0%	20.0%	4.0%	8.9%	100.0%
china oilseed	0.0%	10.0%	4.0%	7.8%	100.0%
cocoa usa	0.0%	0.0%	2.0%	4.0%	90.0%
earn italy	0.0%	10.0%	2.0%	2.7%	62.5%
japan rapeseed	100.0%	60.0%	12.0%	52.5%	100.0%
Medium pre-tagged result set					
Query	P@1	P@10	P@50	MAP	Recall
carcass usa	0.0%	0.0%	2.0%	0.1%	4.5%
china grain	100.0%	90.0%	36.0%	38.8%	100.0%
coffee colombia	100.0%	90.0%	62.0%	84.3%	100.0%
japan yen	0.0%	10.0%	12.0%	15.1%	100.0%
ussr wheat	100.0%	90.0%	48.0%	53.7%	100.0%
Large pre-tagged result set					
Query	P@1	P@10	P@50	MAP	Recall
acq uk	0.0%	0.0%	8.0%	1.7%	28.9%
corn usa	100.0%	60.0%	40.0%	39.7%	94.2%
interest uk	0.0%	30.0%	26.0%	17.1%	94.1%
interest usa	0.0%	0.0%	0.0%	0.8%	37.7%
japan trade	100.0%	40.0%	40.0%	38.0%	98.8%

Table 2: Precision and Recall for the Phrase system

map	p@1	p@10	p@50	recall
Small pre-tagged result set				
2.3%	0.0%	0.0%	-0.4%	0.0%
Medium pre-tagged result set				
-0.1%	0.0%	0.0%	1.2%	0.0%
Larger pre-tagged result set				
3.9%	20.0%	12.0%	12.0%	0.0%

Table 3: Average Precision and Recall Increase

5 Conclusion

This study has proposed a phrase indexing scheme with an evaluation through a vector space search engine. As far as we know, this is the first study that builds the phrase index in the techniques mentioned in the earlier sections of this paper. We have also shown that the phrase indices have shown an advantage in the average precision and recall for queries that have around 100 relevant results, or more, within the Reuters 21578 dataset. We have also looked at the size increase of using the phrase indices, and have seen a factor of 30 increase in the disk usage of the indices. As for the speed deficiency, it was seen as a negligible factor. We have also noted a separate advantage of having such a phrase index, which is the ability of finding suggestions for the system's users. This concept requires further investigation for feasibility and methods of pruning the auto generated suggestions.

References

[1] Linguistic utils: Stop word list. http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words.

[2] Machine learning for natural language processing - esslli 2007 - reuters 21578 xml. http://ronaldo.cs.tcd.ie/esslli07/data/reuters21578-xml/.

[3] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[4] Dirk Bahle, Hugh E. Williams, and Justin Zobel. Efficient phrase querying with an auxiliary index. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 215–221, New York, NY, USA, 2002. ACM.

[5] Matthew Chang and Chung Keung Poon. Efficient phrase querying with common phrase index. *Inf. Process. Manage.*, 44(2):756–769, 2008.

[6] Surajit Chaudhuri, Kenneth Church, Arnd Christian König, and Liying Sui. Heavy-tailed distributions and multi-keyword queries. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 663–670, New York, NY, USA, 2007. ACM.

[7] Mikio Hirabayashi. Benchmark Test of DBM Brothers. http://tokyocabinet.sourceforge.net/benchmark.pdf.

[8] Mikio Hirabayashi. Introduction to Tokyo Products. http://tokyocabinet.sourceforge.net/tokyoproducts.pdf.

[9] Frans Kaashoek, Omprakash D Gnawali, and Omprakash D Gnawali. A keyword set search system for peer-to-peer networks, 2002.

[10] Giridhar Kumaran and James Allan. Adapting information retrieval systems to user queries. *Information Processing & Management*, In Press, Corrected Proof.

[11] Lintao Liu, Kyung Dong Ryu, and Kang-Won Lee. Keyword fusion to support efficient keyword-based search in peer-to-peer file sharing. *Cluster Computing and the Grid, 2004. CCGrid 2004. IEEE International Symposium on*, pages 269–276, April 2004.

[12] Wenlei Mao and Wesley W. Chu. The phrase-based vector space model for automatic retrieval of free-text medical documents. *Data Knowl. Eng.*, 61(1):76–92, 2007.

[13] C.J. van Rijsbergen, S.E. Robertson, and M.F. Porter. New models in probabilistic information retrieval. 1980.

[14] Hugh E. Williams, Justin Zobel, and Dirk Bahle. Fast phrase querying with combined indexes. *ACM Trans. Inf. Syst.*, 22(4):573–594, 2004.

[15] Wei Zhang, Shuang Liu, Clement Yu, Chaojing Sun, Fang Liu, and Weiyi Meng. Recognition and classification of noun phrases in queries for effective retrieval. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 711–720, New York, NY, USA, 2007. ACM.