Journal of Chinese Language and Computing 15 (1): (45-51)

Using Longest Common Subsequence Matching for Chinese Information Retrieval

Yun Xiao[1] Robert W.P.Luk[1] K.F.Wong[2] K.L.Kwok[3]

[1] Department of Computing
The Hong Kong Polytechnic University
Email: {csyxiao, csrluk}@comp.polyu.edu.hk

[2]Department of Systems Engineering & Engineering Management
Chinese University of Hong Kong
Email: kfwong@se.cuhk.edu.hk

[3] Department of Computer Science
Queen's College, CUNY
Email: kwok@ir.cs.qc.edu

Abstract

This paper is about adopting the longest common subsequence (LCS) matching for Chinese information retrieval. We re-ranked the retrieved documents by a mixture of the original similarity score and the LCS score obtained by matching the document titles and the query. This LCS-based similarity score is also used in pseudo-relevance feedback in various ways (e.g., selecting terms and filtering documents with low LCS values). We evaluated the use of LCS in title re-ranking and PRF based on the NTCIR-4 test collection for Chinese ad hoc information retrieval. For title queries, our best MAP achieved is 26.7% evaluated using rigid relevance judgement and 30.2% evaluated using relax relevance judgement.

Keywords

Chinese information retrieval, Longest common subsequence, String matching, Title re-ranking

1 Introduction

Longest common subsequence (LCS) matching (Hirschberg 1977; Apostolico and Guerra 1987; Bergroth et al. 2000; Navarro 2001) is a commonly used technique to measure the similarity between DNA sequences in molecular biology studies (Almeida et al. 2003; Alves

et al. 2003) and between two strings in other research areas like evaluating machine translation (Saggio et al. 2002; Lin 2004). It measures the longest total length of all the matched substrings between two strings (Chin and Poon 1990; Irving and Fraser 1992) where these substrings are allowed to be non-contiguous and these substrings appear in the same order as they appear in the other string. DNA sequences shared many common characteristics with Chinese text in that both types of sequences do not have any space to delimit words and word orders are important for both types of sequences. Given these similarities, we are interested to examine how to use LCS matching for Chinese information retrieval.

Pseudo-relevance feedback (PRF) is one of the most well known (Buckley et al. 1992) and widely applied technique in the open IR evaluation workshops. But sometimes we may be selecting terms from some irrelevant documents. These terms might hurt the retrieval effectiveness. We used LCS to obtain the similarity of the query and the title of the documents to find more relevant documents in the top N retrieved documents.

The rest of the paper is organized as follows. Section 2 discusses the title re-ranking strategy and LCS application in it. Section 3 describes our IR process when introducing LCS to PRF. Section 4 gives the experimental results. Section 5 describes our findings and our future work.

2 Title Re-ranking Strategy

In our IR system, we first retrieve the relevant documents according to the similarity of the query and the documents, and then select the terms in the top N retrieved documents to supplement and enrich the initial query. After that we retrieve the documents the second time for better retrieval effectiveness.

This process does not consider the information in the titles of the documents. Generally speaking, the title of one document gives some clues to the content of this document. For one query, it is plausible that the percentage of the title which has high matching sore with the query in the relevant documents is higher than that in the irrelevant documents. So the retrieval effectiveness would be improved if we can make use of the information in the title of the document. Title re-ranking is a strategy which tries to re-rank the documents in the retrieval process based on the matching score of the title query and the title of the document.

2.1 VSM-based Method

In (Luk and Wong 2002), a VSM-based method is adopted to compute the similarity of the title query and the title of documents. It defines $M(q,t(d_i))$ as the number of the matched bigrams between q and $t(d_i)$. The re-ranking function is:

$$sim'(q,d_i) = (sim(q,d_i) - m) \times M(q_t, t(d_i)) + m \quad (1)$$

where $sim'(q,d_i)$ is the new similarity score, $sim(q,d_i)$ is the original similarity score, m is the minimum original similarity score in the top N retrieved documents, $t(d_i)$ is the title of the document and q_t is the title query. This matching function does not distinguish different word orders directly but implicitly and partially by matching bigrams in the title query and the title of the document. The parameter m ensures that the top N ranked documents remain in the top N of the new ranked list.

2.2 Longest Common Subsequence for Title Re-Ranking

Given sequences x and y, z is the longest common subsequence of them if it is the common subsequence with maximum length. For title re-ranking, we introduce LCS to compute the similarity of the title of the document and the corresponding title query. So

$$M(q_t, t(d_i)) = LCS_{unigram}(q_t, t(d_i)) \quad (2)$$

where q_t is the query, d_i is the *i*-th document, $t(d_i)$ is the title string of the document and $LCS_{unigram}$ is the number of matching Chinese characters between the title of the document and the title query q_t.

$LCS_{unigram}$ just reflects the number of matching characters. For bigram indexing strategy, only using $LCS_{unigram}$ will lose some useful information. Next, we introduce LCS_{string} and LCS_{bigram}, LCS_{string} is the string that comprised of all the matched characters using *LCS* matching. If the matched characters do not appear consecutively in the original string, we separate the characters by adding spaces. These added spaces prevent the incorrect subsequent identification of bigrams that never appeared in the original string. LCS_{bigram} is the number of the matched bigrams between the query and LCS_{string}. Finally, we combine $LCS_{unigram}$ and LCS_{bigram} together as follows:

$$M(q_t, t(d_i)) = \beta * LCS_{unigram}(q_t, t(d_i)) + (1-\beta) LCS_{bigram}(q_t, LCS_{string}(q_t, t(d_i))) \quad (3)$$

Table 1 shows the retrieval effectiveness of bigram indexing with title re-ranking when not following PRF. We used the NTCIR-4 test collections for Chinese ad hoc retrieval. The performance of *LCS* is better than that of VSM-based method, which proves that *LCS* can find more matched information than VSM-based method does.

Top 10 documents	Rigid(%)		Relax(%)	
	MAP	P@10	MAP	P@10
VSM-based method	20.40	24.80	23.60	31.40
T-L1 ($LCS_{unigram}$)	20.90	24.92	23.86	31.86
T-L2 (LCS_{bigram})	20.38	24.75	23.60	31.36
T-L12 (Both where β =	21.05	24.92	23.84	31.86

Table 1: Retrieval effectiveness of bigram indexing with title re-ranking (for the top 10 retrieved documents)

3 Integrating PRF and Title Re-ranking

In (Luk and Wong 2002), PRF and title re-ranking were combined to obtain even better performance. Since LCS seemed to be a promising matching technique than VSM-based method, we examine how LCS can be used in title re-ranking and PRF together. Specifically, figure 1 shows the entire process of our IR system. In the figure, S0, S1, S2, S3, S4, S5, S6 are the switches of our system; *A* and *B* are the different states in each switch. If the title of a document has a high LCS value with the title query, we call this document high LCS value document.

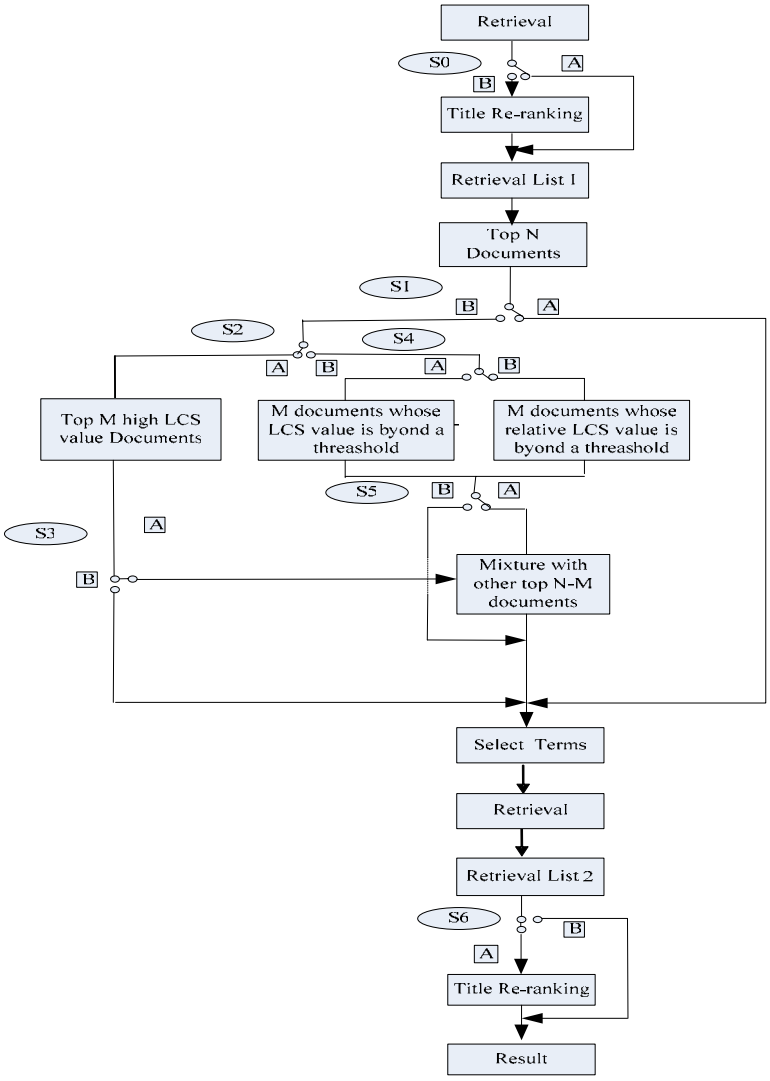

Figure 1: the entire process of our IR system

In normal PRF, it is assumed that the top *N* documents are relevant to the query. Of course this is not always true. Since selecting terms from the irrelevant documents is likely to hurt the retrieval performance, we try to distinguish more relevant documents from the top *N* retrieved documents using LCS matching between the titles of the documents and the query.

For switch S0, state *A* and *B* denote whether we adopt title re-ranking strategy before PRF. For switch S1, *A* selects terms from the top *N* documents directly. In switch S2, *A* selects terms from top *M* documents which have high LCS values in the top *N* retrieved documents. For switch S4, *A* selects terms from the documents whose LCS values are beyond a threshold α in the top *N* retrieved documents. Because for a long query, the LCS value may be higher than a short query, then it is not easy to set a threshold for all queries. So we use another LCS

value, relative LCS value, which is the LCS value divided by the length of the query. State *B* selects terms from the documents whose relative LCS values are beyond β in the top *N* retrieved documents.

The above schemes only select some high LCS value documents and discard some low LCS value documents. But the LCS values of some relevant documents may be very low, so it is not reasonable to not consider the low LCS value documents in the top *N* retrieved documents. We tried to consider the high LCS value documents and other low LCS value documents together. Our term selection method for PRF is based on the term frequency and the inverse document frequency in the top *N* documents. Then term frequency in the high LCS value documents is re-weighted as follows:

$$tf_{new}(\delta) = tf(\delta) + \lambda \quad (4)$$

where $tf(\delta)$ is the original term frequency of the term δ in the high LCS value documents, $tf_{new}(\delta)$ is the new term frequency of the term, λ is the value to be used to "boost" the term frequency. In switch S3 and S5, state *A* mixes high LCS value documents with other documents in top *N* retrieved documents.

4 Evaluation

Our evaluation is based on NTCIR 4 data set. We only used title queries in this evaluation. For comparison with our results in (Luk and Wong 2002), our runs are based on bigram indexing as in (Luk and Wong 2002).

4.1 Results

Table 2 shows the retrieval performance of our system. Because the performance of PRF followed by title re-ranking is better than the other orders of applying PRF and re-ranking, we always select state *A* in switch S0 and S6.For clarity, we reported the best result of each scheme. Scheme 0 is the method mentioned in (Luk and Wong 2002) but with a different parameter setting. Specifically, we obtained the best performance when selecting 75 terms from top 7 documents for PRF and using top 2000 documents for title re-ranking. We used the same number of the selected terms and the documents for title re-ranking.

Scheme	Switches					Parameters	Rigid(%)		Relax(%)	
	S1	S2	S3	S4	S5		MAP	P@10	MAP	P@10
0	A	-	-	-	-	N=7	26.41	**29.83**	30.15	**38.31**
1	B	A	B	-	-	N=7,M=6	25.42	28.14	29.41	37.29
2	B	B	-	A	B	N=7, α=1	24.95	28.14	29.18	37.29
3	B	B	-	B	B	N=7, α =0.01	25.10	28.31	29.37	37.29
4	B	A	A	-	-	N=7,M=4,λ =2	**26.70**	28.92	28.45	37.57
5	B	B	-	A	A	N=8,α =2,λ =2	26.24	28.64	30.19	37.46
6	B	B	-	B	A	N=7,α=0.25,λ=2	26.44	28.98	**30.27**	37.80

Table 2: Performance of different schemes using LCS matching in Chinese IR

For all schemes, we obtained the best result when *N* is 7 except scheme 5. Since the best performance of scheme 1, 2 and 3 are worse than scheme 0, some relevant documents have low LCS values and just selecting the high LCS value documents will discard some relevant documents. The best performances of scheme 4, 5 and 6 are better than scheme 1, 2 and 3, which showed that the mixture method is more effective. In scheme 4, we obtained our best rigid mean average precision (MAP), 26.7%, which improve as much as 1.8% than the best rigid MAP 24.9% in (Luk and Wong 2002). It is believable that when we give a high weight to term frequency of the terms from high LCS value documents, more relevant terms of the query terms have been selected for PRF. We can see the best rigid MAP is achieved by scheme 4 and the best relax MAP is achieved by scheme 6, so the rigid MAP and relax MAP are two different evaluation methods.

5 Conclusion and further work

Title re-ranking is an effective strategy to re-ranking the documents in the retrieval list. We introduce the longest common subsequence (LCS) matching to our title re-ranking strategy and it is shown to be marginally better than our VSM-based method which just used the number of matched bigrams to compute the similarity of the title query and the title of the document.

For PRF, LCS can help to distinguish relevant documents from the top *N* documents. We have tried some schemes and found that mixing the high LCS documents and other documents will get a better performance. If the term frequency of a term that has a high LCS value in a document is given a high weighting, we can get the best rigid MAP as high as 26.7%. Our future work will examine the use of LCS matching in title re-ranking and PRF for different data sets such as NTCIR-3. We will use this method for other query types to see if it can improve the retrieval performance further.

Acknowledgements

This work is supported by the CUHK Strategic Grant (#4410001) and by the Hong Kong Polytechnic University Grant number: PolyU 5183/03E.

References

A. Apostolico, C. Guerra. 1987. The Longest Common Subsequence Problem Revisited, *Algorithmica*, pp. 315-336.

C. Buckley, G. Salton, J. Allan. 1992. Automatic retrieval with locality information using Smart, *the 1st Proceedings of Text Retrieval Conference*, pp. 59-72.

C. E. R. Alves, E. N. Cáceres, S. W. Song. 2003. A Parallel Application in Grid Computing for the Longest Common Subsequence, *International Conference on Bioinformatics and Computational Biology* (ICoBiCoBi 2003), Ribeirao Preto, Brazil, May pp. 14-16.

C.Y Lin. 2004. Looking for a Few Good Metrics, *Working Notes of NII-NACSIS Test Collection for IR Systems* (NTCIR-4) Tokyo.

D. S. Hirschberg. 1977. Algorithms for the longest common subsequence problem, *Journal of the ACM*, Vol. 24, No. 4, pp. 664–675.

F. Y. L. Chin, C. K. Poon. 1990. A fast algorithm for computing longest common subsequences of small alphabet size, *Journal of Information Processing*, v.13 n.4, pp. 463-46.9

G. Navarro. 2001. A Guided Tour to Approximate String Matching, *ACM Computing Survey*, Vol. 33, No. 1, pp. 31-88.

H. Saggio, D. Radev, S. Teufel and W. Lam. 2002. Meta-Evaluation of Summaries in a Cross-Lingual Environment Using Content-Based Metrics, *19th International Conference on Computational Linguistics (COLING-2002)*, Taipei, Taiwan.

L. Bergroth, H. Hakonen, T. Raita. 2000. A survey of longest common subsequence algorithms, *7th International Symposium on String Processing Information Retrieval*, pp. 39–48.

N.F. Almeida Jr, C. E. R. Alves, E. N. Cáceres and S. W. Song. 2003. Comparison of Genomes using High-Performance Parallel Computing, *Proceeding of the 15th Symposium on Computer Architecture and High Performance Computing* - SBAC-PAD 2003, Brazillian Computer Society (SBC), Sao Paulo - SP- Brazil, November 10-12, IEEE Computer Society, pp. 142-148.

R.W. Irving, C.B. Fraser. 1992. Two algorithms for the longest common subsequence of three (or more) strings, *Proceedings of the 3rd Annual Symposium on Combinatorial Pattern Matching*, LNCS 644: pp. 214-229.

R. W.P. Luk, K.F. Wong. 2004. Pseudo-Relevance Feedback and Title Re-ranking for Chinese Information Retrieval, *Working Notes of NII-NACSIS Test Collection for IR Systems* (NTCIR-4) Tokyo.