Investigating Sentence Weighting Components for Automatic Summarisation

Shao Fen Liang*, Siobhan Devlin, John Tait

School of Computing and Technology, University of Sunderland, Sunderland, SR6 0DD, UK

Abstract

The work described here initially formed part of a triangulation exercise to establish the effectiveness of the Query Term Order algorithm. The methodology produced subsequently proved to be a reliable indicator of quality for summarising English web documents. We utilised the human summaries from the Document Understanding Conference data, and generated queries automatically for testing the QTO algorithm. Six sentence weighting schemes that made use of Query Term Frequency and QTO were constructed to produce system summaries, and this paper explains the process of combining and balancing the weighting components. We also examined the five automatically generated query terms in their different permutations to check if the automatic generation of query terms resulting bias. The summaries produced were evaluated by the ROUGE-1 metric, and the results showed that using QTO in a weighting combination resulted in the best performance. We also found that using a combination of more weighting components always produced improved performance compared to any single weighting component.

Keywords: Query Term Order; Query Term Frequency; Sentence Location; Sentence Order; Sentence Weighting Scheme

1. Introduction

Sentence based summarisation techniques are commonly used in automatic summarisation to produce extractive summaries (Yeh et al., 2005; Guo and Stylios, 2005). The techniques first break a document into a list of sentences. Important sentences are then detected by some sentence weighting scheme, and the highly weighted sentences are selected to form a summary. Although researchers know that the sentence extraction techniques often result in summaries that lack coherence, the generated summaries are useful for humans to browse (Paice & Jones, 1993; Hirao et al., 2002) and make judgements about.

A sentence weighting scheme can be variously formulated by employing many components and distributing them with different parameters. For example, Term Frequency, Sentence Order and Sentence Length are common components. However, the detail of how to formulate a sentence weighting scheme is rarely discussed and reported in the literature. This misty area could be cleared by conducting several experiments to show the importance of sentence weighting scheme in automatic summarisation. Furthermore, automatic summarisation systems that are suited for our purpose are not readily available. Therefore in this paper, we conduct our own experiment and focus on investigating and comparing effectiveness between Query Term Frequency (QTF) and Query Term Order (QTO), and evaluating the summaries produced with the ROUGE-1 metric. QTF in the rest of our sentence weighting algorithm means the number of times the query terms appear in a sentence, and each term is equally weighted. QTO means the number of times the query terms appear in a sentence, with those terms appearing earlier in the query being assigned higher scores than those appearing later. By comparing QTO with QTF we should be able to discover if order is important for query biased summarisation when users construct queries.

Although using QTO alone can improve search result summaries, we are interested in further investigating whether involving QTO in different combination of sentence weighting scheme would work better than QTO solely. The decision to use DUC data for this experiment instead of repeating the previous work was due to concern over time and effort of the human subjects.

* Corresponding author. Tel.: +44 191 515 3410; Fax: +44 191 515 3461.
Email address: ShaoFen.Liang@sunderland.ac.uk, Siobhan.Devlin@sunderland.ac.uk, John.Tait@sunderland.ac.uk.

2. Sentence extraction, and query terms, and query length

The early work from Luhn (1958) identified that a significant sentence consisted of a set of significant words. The definition of significant words in his work avoided linguistic implications such as syntax but gave a statistical table of *total different words*, *less different common words*, *different non-common words* and their occurrences. The words in Luhn's work are every single word in a document without any pre-processing (e.g. stemming). In 1969, Edmundson pointed out four distinctive term types namely *cue*, *key*, *title* and *location*. These four term types derive four methods for extracting summaries, and also proved that terms contain important clues for producing summaries.

Since people began to frequently search information online, the relationship between terms in a query and documents has become an active research area. Robertson (1990) discussed using term weighting to generate new terms and examine the usefulness of the new terms as a query explanation approach. Tombros and Sanderson (1998) proved that users could better judge the relevance of documents if their query terms appeared in the summaries. Manabu and Hajime (2000) combined the use of query terms and lexical chains to produce query-biased summaries. White et al. (2003) used a combination of query terms, Edmundson's title and location to determine important sentences.

Several studies about query length from 1981 to 1997 (Fenichel, 1981; Hsieh-yee, 1993; Bates et al., 1993; Spink & Saracevic, 1997) with novices, moderately experienced and experienced searchers, searchers who were familiar with the search topics and those who were not, and humanities scholars have come to the conclusion that an average query length was in the range of 7-15 terms. Jansen et al.'s (2000) studied query length by using search engine log. Their studies indicated that the length of a real query from real users was on average 2.21 terms from the range of 0 to 10 terms, and also that query length declined from 1981 to 2000. This result was an inspiration for our proposed Query Term Order algorithm. However we decided to use the top 5 frequent terms in our experiment, reflecting the more recent work of Williams et al. (2004), who selected phrase length from 2 to 7. Five, therefore seemed a reasonable length to use.

3. Query Term Order examination with DUC

Evidence that automatic summarisation is improved by the use of Term Order in both documents and queries has been reported in our previous work (Liang, 2005). The central idea of the Query Term Order algorithm is to pay attention to the order of a user's query terms. As the previous research showed that query length is generally short, processing the QTO algorithm for online summarisation is not complex and can generate a set of weighting terms from the input query terms to enhance weighting effectiveness. Although our proposed Query Term Order algorithm proved effective for producing search result summaries with English web documents (Liang, 2006), we wished to triangulate the study to establish the algorithm's effectiveness using different sets of data.

Document Understand Conference (DUC, 2004) data was used for this experiment. The data originated from task 1 of the competition in DUC 2004. The task was to produce very short single-document summaries. The length of the produced summaries was restricted to no more than 75 bytes (which is about one line of an typical A4 sheet) including spaces and punctuations. The data contains 50 English Newswire clusters, each with 10 documents of a similar content. After the competition, DUC asked 8 human abstractors to write summaries for the 500 documents. These people produced 8 sets of summaries as the gold standard summaries for evaluating participants' systems. We utilised these gold standard summaries for comparison with our system produced summaries.

Lack of queries for the QTO algorithm was the first problem that we encountered. Therefore we generated our own queries in order to produce summaries. Term Frequency (TF) was employed to generate queries as it is one of the most common techniques used in automatic query generation (Somlo & Howe, 2003). A list of 235 stopwords was removed from the documents and no stemming technique was used. The stopword list is slightly modified from the stopword list of the Glasgow University information retrieval research group. (Glasgow University, 2006). Words relating to date or part of a day were also removed, such as *Sunday, Monday, Sun, Mon, morning, afternoon* and so on. The top five frequent terms from each cluster were selected as a query, so that 50 queries were generated for the 50 DUC clusters.

Each query generated 10 summaries: one summary for each document in the cluster. There were 50 clusters, and we used six sentence weighting schemes (see section 4), resulting in a total of 3,000 summaries used in our experiments. We kept our summary length to the 75 character limit imposed by DUC, in order to be the same length as the human summaries for the ROUGE evaluation.

In addition, using automatic generation of queries may result too artificial to be biased of our proposed QTO algorithm. Thus, we isolated the comparison to between QTO and QTF by using different order permutation among the 5 automatically generated query terms (see section 5).

4. Six sentence weighting schemes

We focused on investigating four sentence weighting components namely: Query Term Order (QTO), Query Term Frequency (QTF), Sentence Length (SL) and Sentence Order (SO).

The most important idea in the QTO algorithm is that however a query is processed the order in the original query is preserved all the time. Formula (1) shows how the QTO score is calculated, where s_1, s_2, s_3 and s_j represent a number of j segmentations respectively. The segmentations are derived by removing stop words from an input query and taking a sequence of contiguous words between either punctuation or a stop word as a segment. Although stop words were omitted after the first split from the original query, the order existing between $s_1 \ldots s_j$ is the same as the order in the original query. Each of the segmentations has a second split into some single terms. The second split may be unnecessary if the segmentation already contains a single term only. Therefore t_1, t_2, t_3 … t_k represents terms from second split, and f_1, f_2, f_3 … f_m represents the frequencies of QTO's weighting terms in a sentence respectively. Each weighting term is assigned a score in descending order (i.e. s_1 is assigned $j+k$, s_2 is $j+k-1$ … and t_k is 1.). Therefore the QTO score of each sentence is $f_1*(j+k)+f_2*(j+k-1)\ldots.+f_m*1$. The j and k are unlikely to be equal because j is the liner order position of the segment, and k is the position of the term within the segment. The m is the total number of weighting terms, therefore it is equal to j+k.

$$QTO = [s_1 \quad s_2 \quad s_3 \ldots s_j \quad t_1 \quad t_2 \quad t_3 \ldots t_k] \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ . \\ . \\ . \\ f_m \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

QTF is used to calculate the frequency of each query term in a sentence. Formula (2) represents how QTF is calculated, where t_1, t_2, t_3 … t_n represents terms in a query, and f_1, f_2, f_3 … f_n represents term frequency of t_1, t_2, t_3 … t_n respectively. Each of t_1, t_2, t_3 … t_n were equally assigned 1. Therefore each sentence's QTF score is $f_1 + f_2 + f_3 \ldots + f_n$.

$$QTF = [t_1 \quad t_2 \quad t_3 \quad ..t_n] \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ . \\ . \\ f_n \end{bmatrix} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The Sentence Length (SL) score is shown in formula (3). Each sentence's length is calculated according to how many spaces (x) are in the sentence. For example if $x=1$ then the $SL = 2$, which means the sentence contains 2 words.

$$SL = x + 1; x = \{1, \quad 2, \quad 3, \quad ...\} \quad \text{……..……… (3)}$$

Sentence Order is scored in descending order as shown in formula (4), where y represents the scores. Therefore the earliest sentence is scored highest and the latest sentence is scored 1.

$$SO = y; y = \{..., \quad 3, \quad 2, \quad 1\} \quad \text{……..……… (4)}$$

We produced six summarisers for the experiment. They are named A, B, C, D, E and F and described in the following section. In addition, we adjusted parameters – in C and F - in order to discover the best combination for the weighting scheme. Following weighting, we also tested omitting short sentences with different thresholds from 4 to 10 words (Kupiec et al., 1995). The reason to add threshold as one variable is to check if threshold affecting summary result after weighting procedure in automatic summarisation.

A. QTO : The single component QTO is used in the A weighting scheme. We do not give any parameter to adjust the QTO score because it is independent without any combination.
B. QTO/SL : We considered using sentence length to balance the QTO score in case of a longer sentence more easily scoring higher than a shorter sentence. We assumed that the way we calculated the B scheme was fair in application to every sentence, so we did not use any parameter to adjust the result scores.
C. $(\alpha)(QTO/SL) + (\beta)SO$: SO was included to expand scheme B into a combination of two components (i.e. QTO/SL and SO). There is a problem with this combination because we do not know if SO has a greater chance of dominating the scheme or the other way around. For example, there are five terms in each query in our experiment, but there may be 50 or more sentences in a document. SO will always score between 1 and 50 but the QTO/SL has a very low chance of scoring higher than 5. Even when they are both normalised to between 0 and 1, the intervals of QTO/SL and SO are different, in that there are only 5 possible points on the query terms scale yet there are 50 possible points on the scale of sentences in a document. Therefore the scale with the largest intervals will dominate the combination of QTO/SL. Thus, we needed to find the best parameter distribution of the combination. Different ratios of α : β were tried as shown in Table 2.
D. QTF : This scheme is used as a comparison with QTO. Each term appearing in a query is treated the same, and a sentence's QTF score is calculated according to the frequency of the query terms in formula (2). The reason for not using a parameter to adjust the result score is the same as for scheme A.
E. QTF/SL : The scheme is used for comparison with the B scheme, and constructed for the same reason as B.
F. $(\alpha)(QTF/SL) + (\beta)SO$: This is also used to compare with C.

5. Different order permutation

The five automatically generated terms were placed in different sequences in order to investigate if the QTO algorithm outperforms QTF in various term orders. They are named *Highest, Reverse, Random* and *Verbatim*. Table 1 shows example queries of the four different permutations. These four different queries are for DUC2004 d30001t cluster. The top five frequent terms in *Highest* are placed as the most frequent term first then the second until the fifth. *Reverse* is to reverse the term order for the terms in *Highest*, therefore the first term in *Highest* is placed the last in *Reverse* and the last in *Highest* is placed the first in *Reverse*. *Random* is a random order generated from the five terms in *Highest*. *Verbatim* uses a quotation from the document cluster as the query. Ideally this will be a usage of exactly the five automatically generated terms, but if there is no such usage a four term quotation will be selected, and so on. Where there are several alternative verbatim term order quotations of the same length in the document cluster the most frequently occurring one is selected.

Table 1

Different order permutations and their descriptions of the automatic generated queries.

Order Name	Description of the order	Query of DUC2004 d30001t cluster
Highest	Highest frequent word first	hun sen ranariddh said party
Reverse	Reverse order from Highest	party said ranariddh sen hun
Random	Random order from Highest	ranariddh sen said hun party
Verbatim	Verbatim order from Highest	hun sen said party ranariddh

6. Evaluation with ROUGE

To evaluate our 6*500 summaries produced from the different sentence weighting schemes, we employed the ROUGE metric (Lin, 2004). Although ROUGE contains many metrics, we only used ROUGE-1 for the evaluation. There are two reasons for the decision. The first one is that ROUGE is an extended version of BLEU, and Papineni et al. (2000) indicated that the unigram precision yields a score which more closely matches human judgements. Also n-gram precision decays roughly exponentially with n in their experiment. The second reason, illustrated in Figure 1, is that DUC 2004 ROUGE evaluation is similar to Panpineni's report. The legends H1 to H8 in Figure 1 are 8 sets of human produced summaries. These were scored by using one set of summaries as system summaries, and the other 7 sets as gold standard summaries. ROUGE-1 then computed simulated system summary scores. ROUGE evaluations show that ROUGE-1 has the highest scores. The scores decline roughly exponentially when the n-gram increases. Even though ROUGE contains N-gram, Longest Common Subsequence and Weighted Longest Common Subsequence metrics, ROUGE-1 (unigram) effectively predicts system ranking based on the other scores. The ROUGE-1 is to use each word in the system summary to compare with the eight gold standard summaries to calculate its recall. The computed 8 recall were then be averaged as the result.

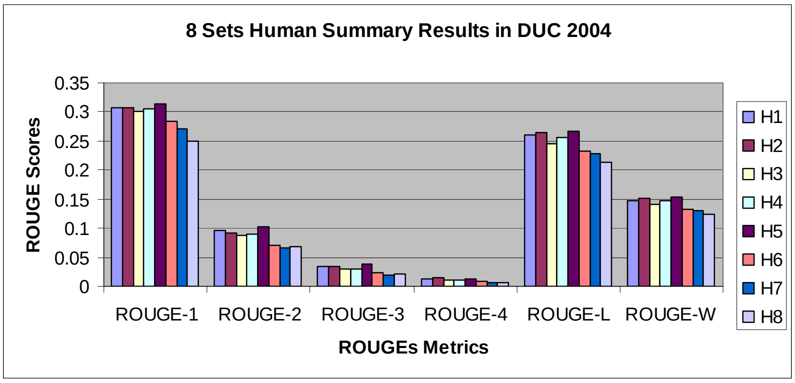

Fig. 1. DUC 2004 ROUGE scores of human summaries

Table 2 shows ROUGE-1 evaluation results of the C scheme in each entry cell, where the first left column shows the α parameter increases from 0.1 to 0.9 while β decreases from 0.9 to 0.1. The top row shows the threshold of each sentence is from 4 words long to 10. The comparison graph is shown in Figure 2, where the 3:7 ratio is the highest and 1:9 is the lowest among the 9 different α and β ratios in the C scheme.

Table 2

ROUGE-1 evaluation results for different parameter distribution in the C scheme

α : β	4	5	6	7	8	9	10
1:9	0.0515	0.0516	0.0510	0.0516	0.0513	0.0527	0.0520
2:8	0.0532	0.0535	0.0528	0.0533	0.0531	0.0538	0.0532
3:7	0.0536	0.0540	0.0533	0.0538	0.0535	0.0541	0.0534
4:6	0.0533	0.0536	0.0529	0.0534	0.0532	0.0537	0.0531
5:5	0.0528	0.0533	0.0526	0.0531	0.0529	0.0534	0.0533
6:4	0.0531	0.0536	0.0530	0.0535	0.0532	0.0536	0.0535
7:3	0.0531	0.0536	0.0530	0.0536	0.0534	0.0537	0.0536
8:2	0.0528	0.0532	0.0526	0.0532	0.0530	0.0533	0.0531
9:1	0.0527	0.0530	0.0523	0.0529	0.0527	0.0530	0.0529

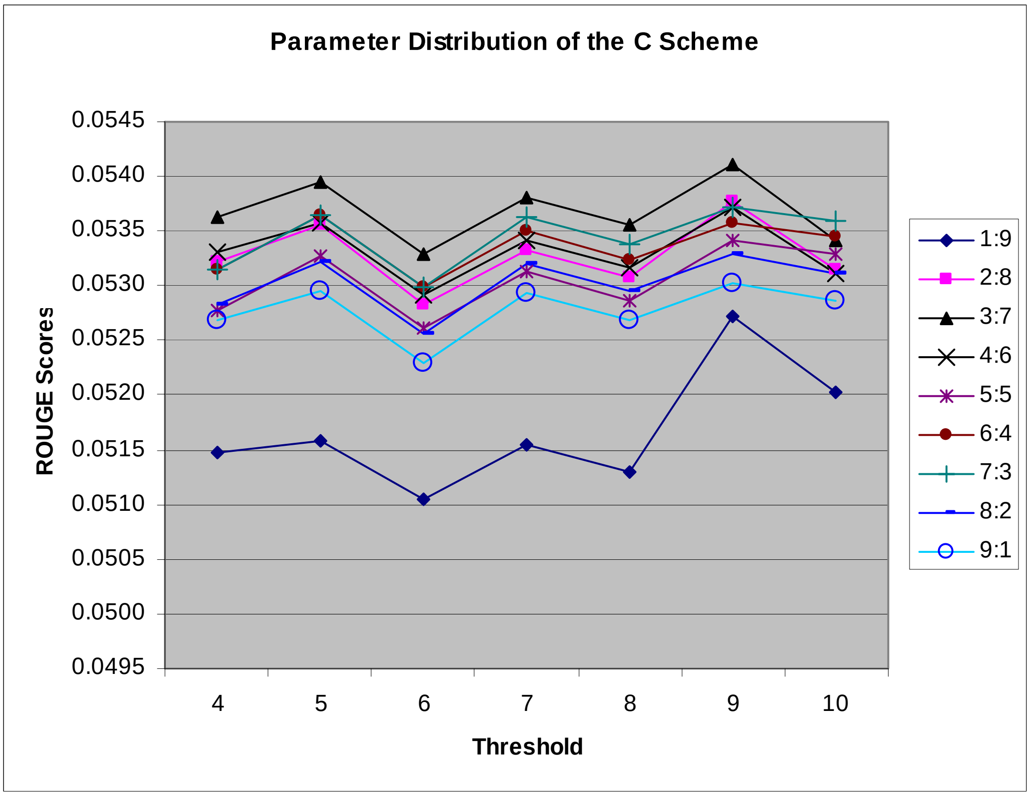

Fig. 2. Parameter distribution of the C scheme

Table 3 shows ROUGE-1 evaluation results of the F scheme. The table structure is the same as Table 2. Their results are compared in Figure 3, where the 4:6 ratio is the highest and 1:9 is still the lowest one among the 9 different α and β ratios in the F scheme. The two different highest ratios shown in tables 3 and 4 are different, therefore we cannot conclude a single best α and β as the best parameter for the C and F schemes for any corpus. This is the case for DUC 2004 data only.

Table 3

ROUGE-1 evaluation results for different parameter distribution in the F scheme

α : β	4	5	6	7	8	9	10
1:9	0.0503	0.0500	0.0500	0.0511	0.0510	0.0510	0.0511
2:8	0.0509	0.0512	0.0513	0.0521	0.0517	0.0518	0.0514
3:7	0.0500	0.0507	0.0508	0.0516	0.0516	0.0520	0.0522
4:6	0.0512	0.0516	0.0518	0.0526	0.0528	0.0530	0.0533
5:5	0.0507	0.0512	0.0513	0.0520	0.0523	0.0526	0.0528
6:4	0.0507	0.0512	0.0513	0.0520	0.0522	0.0527	0.0529
7:3	0.0502	0.0505	0.0507	0.0514	0.0516	0.0520	0.0524
8:2	0.0503	0.0506	0.0507	0.0515	0.0518	0.0523	0.0525
9:1	0.0502	0.0508	0.0509	0.0516	0.0520	0.0524	0.0527

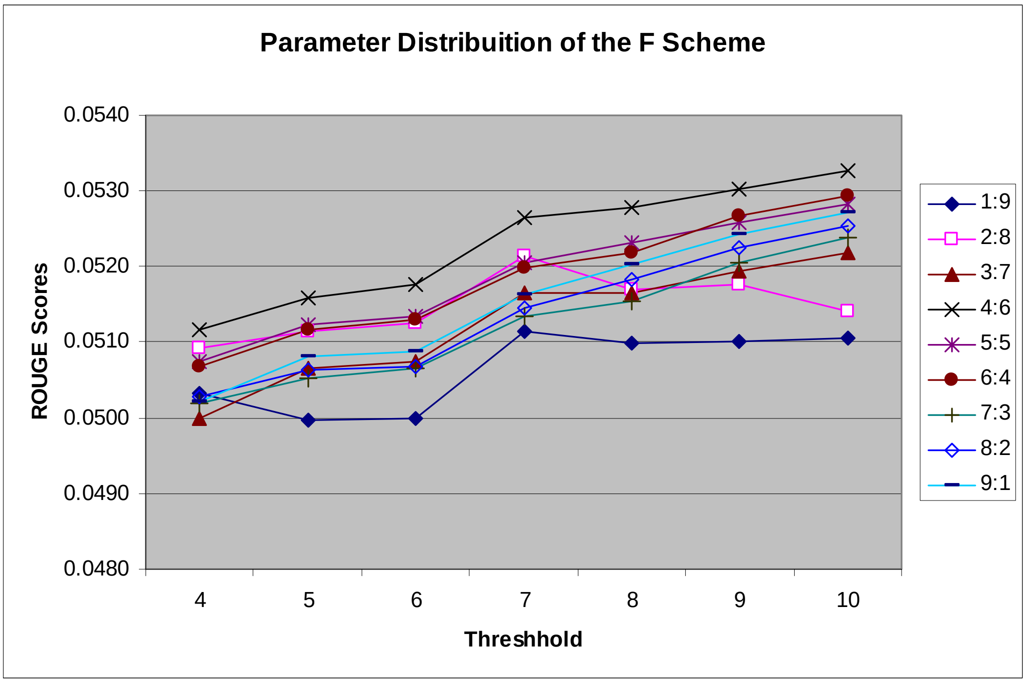

Fig. 3. Parameter distribution of the F scheme

Table 4 shows the results of all 6 weighting schemes, where the results for C and F are the highest parameter ratios taken from tables 2 and 3 respectively. Fig. 4 shows ROUGE-1 evaluation results, and clearly demonstrates that using a single weighting component (i.e. *A* and *D*) achieved the worst results. Although the results show that *A* is slightly worse than *D*, we can only assume that the use of a term frequency algorithm to generate queries automatically has already given the advantage to Query Term Frequency (the *D* scheme). However, the C scheme performed the best, and in addition, using QTO in a combination performed better than without. For example, *B* clearly shows better results than *E*, and *C* is also better than *F*. We can be almost certain that QTO performs better than QTF. If we group the six weighting schemes into (A, B, C) and (D, E, F) we find that a combination with more weighting components always performs better than fewer (i.e. C>B>A and F>E>D). In this experiment, threshold does not have any significant impact on the results.

Table 4

ROUGE-1 evaluation results of A-F with threshold from 4 to 10

	4	5	6	7	8	9	10
A (QTO)	0.0471	0.0471	0.0472	0.0472	0.0473	0.0466	0.0466
B (QTO/SL)	0.0522	0.0522	0.0518	0.0524	0.0521	0.0525	0.0522
C (0.3)QTO/SL+(0.7)SO	0.0536	0.0540	0.0533	0.0538	0.0535	0.0541	0.0534
D (QTF)	0.0475	0.0475	0.0474	0.0474	0.0475	0.0473	0.0474
E (QTF/SL)	0.0492	0.0497	0.0497	0.0502	0.0510	0.0515	0.0517
F (0.4)QTF/SL+(0.6)SO	0.0512	0.0516	0.0518	0.0526	0.0528	0.0530	0.0533

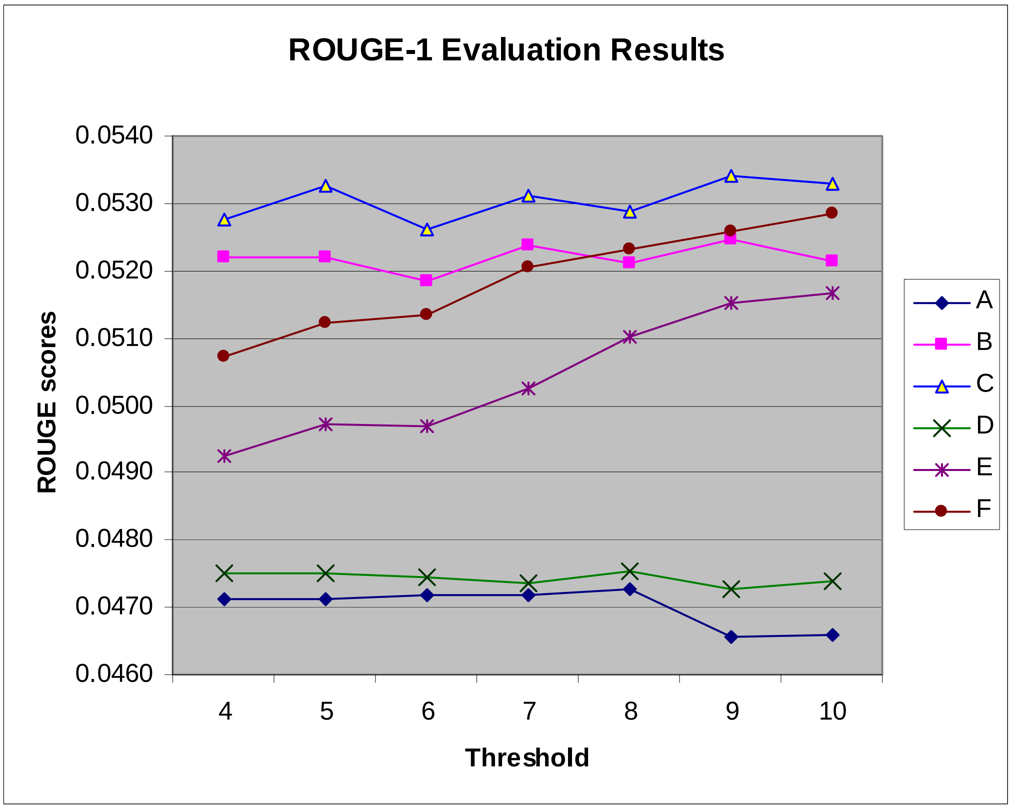

Fig. 4. ROUGE-1 evaluation results in graph

The query term order permutation results between QTO and QTF are shown in Table 5. The different order comparisons are shown in figures 5 to 8 respectively. Three of the four figures (6, 7 and 8) show that QTO performs better than QTF in the orders of *Reverse*, *Random* and *Verbatim*. The only exception is the *Highest* order in Figure 5, which proves our assumption that using top frequent terms as the query has given the advantage to QTF. Figure 9 shows the synthesis results of all order permutation. Each result appears in a similar result when the threshold was chosen to between 4 and 7. It is hard to judge that which of the *Random* and *Reverse* orders of QTF performs the worst among the eight results. However, the *Reverse* order is the worse among the four orders. On the other hand, without the exceptional case of the *Highest* order, the *verbatim* order has performed the best. This result leads a further work to construct a new algorithm of combining QTO and query term verbatim order, which may produce search result summary more effectively.

Table 5
ROUGE-1 evaluation results of QTO and QTF in four different orders with threshold from 4 to 10

QTO vs QTF	3	4	5	6	7	8	9
QTO-Highest	0.0471	0.0471	0.0472	0.0472	0.0473	0.0466	0.0466
QTO-Reverse	0.0464	0.0458	0.0458	0.0459	0.0459	0.0459	0.0456
QTO-Random	0.0470	0.0468	0.0468	0.0468	0.0468	0.0465	0.0464
QTO-Verbatim	0.0485	0.0472	0.0473	0.0473	0.0471	0.0471	0.0472
QTF- Highest	0.0475	0.0475	0.0474	0.0474	0.0475	0.0473	0.0474
QTF- Reverse	0.0460	0.0456	0.0456	0.0456	0.0456	0.0457	0.0455
QTF- Random	0.0454	0.0457	0.0457	0.0457	0.0457	0.0454	0.0455
QTF- Verbatim	0.0481	0.0471	0.0470	0.0470	0.0468	0.0471	0.0471

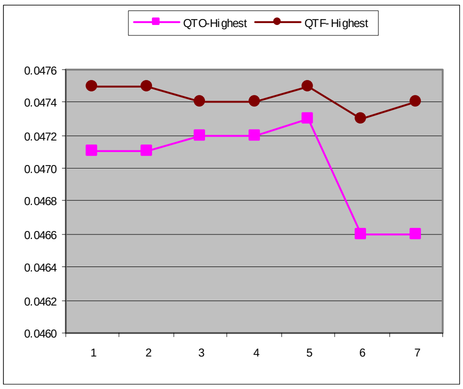

Fig. 5 Highest order comparison

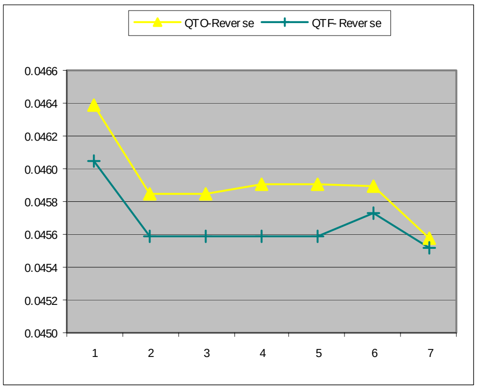

Fig. 6 Reverse order comparison

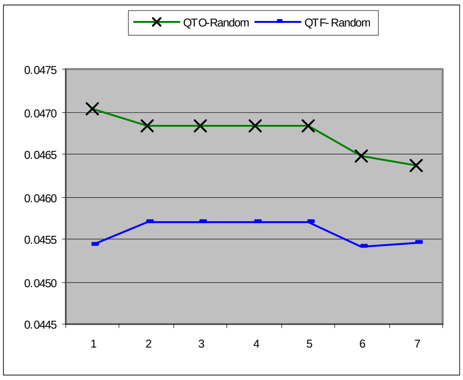

Fig. 7 Random order comparison

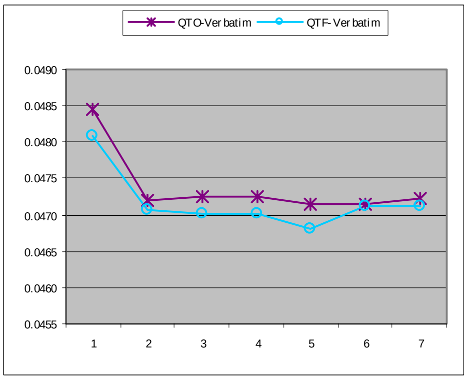

Fig. 8 Verbatim order comparison

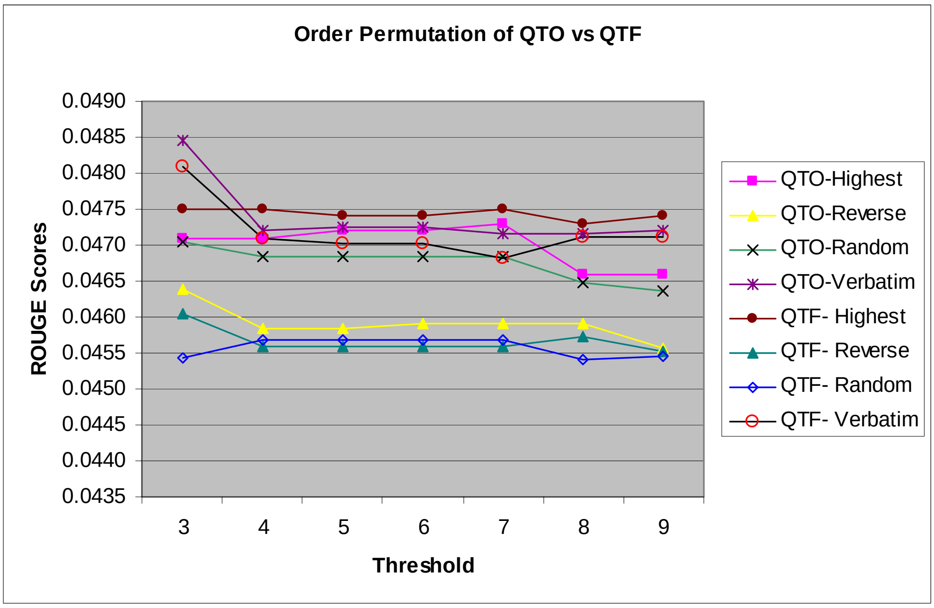

Fig. 9 The four permutations comparison between QTO and QTF

7. Conclusion

In this paper we have examined the importance of the term order in a given query by comparing different sentence weighting schemes for automatic summarisation. The human summaries provided by DUC 2004 were utilised as the gold standard summaries, and compared with system produced summaries. We constructed six weighting schemes and explained how we adjusted them to avoid imbalanced weighting results in producing summaries. The results were evaluated by the ROUGE-1 metric, and show that using a single component in a weighting scheme yields the worst performance. But using QTO in a combination produced promising results. In particular the *C* (0.3QTO/SL+0.7SO) weighting scheme which combines QTO with Sentence Length and Sentence Order performed the best among the six. Finally, Query Term Frequency (QTF) was shown to be the least useful weighting component.

References

Bates, M. J., Wilde, D. N. & Siegfried, S. (1993). An analysis of search terminology used by humanities scholars: The Getty online searching project report. Library Quarterly, 63(1), 1-39.

DUC. (2004). Document Understand Conference. http://www-nlpir.nist.gov/projects/duc/guidelines/2004.html.

Edmundson, H.P. (1969). New methods in automatic extracting. Journal of the Association for Computing Machinery, 16, 2, 264-285.

Fenichel, C. H. (1981). Online searching: Measures that discriminate among users with different types of experience. Journal of the American Society for Information Science, 32, 23-32.

Glasgow University (2006). http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

Guo, Y. & Stylios, G. (2005). An intelligent summarisation system based on cognitive psychology. Information Sciences, 174(1-2), 1-36.

Hirao, T., Isozaki, H., Maeda, E. & Matsumoto, Y. (2002). Extracting important sentences with support vector machines. Proceedings of the 19th international conference on Computational linguistics, 1, 1-7.

Hsieh-yee, I. (1993). Effects of search experience and subject knowledge on the search tactics of novice and experienced searchers. Journal of the American Society for Information Science, 44(3), 161-174.

Jansen, B.J., Spink, A. & Saracevic, T. (2000). Real life, real users, and read needs: a study and analysis of user queries on the web. Information Processing and Management, 36(2), 207-227.

Kupiec, J., Pedersen, J., Chen, & F. (1995). A trainable document summariser. In Proceedings of the 18th annual international conference on research and development in information retrieval (SIGIR'95), 68–73.

Liang, S.F., Devlin, S. & Tait, J. (2005). Using query term order for result summarisation. ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'05, Brazil, 2005, 629-630.

Liang, S.F., Devlin, S. & Tait, J. (2006). Evaluating Web Search Result Summaries. The 28th European Conference

on Information Retrieval (accepted).
Lin, C.Y. (2004). ROUGE: a Package for Automatic Evaluation of Summaries. Proceedings of the Workshop on Text Summarization Branches Out, Barcelona, Spain, 25-26.
Luhn, H.P. (1958). The Automatic creation of literature abstracts. IBM Journal (April, 1958), 159-165.
Manabu, O. & Hajime, M. (2000). Query-biased summarisation based on lexical chaining. Computational Intelligence, 16, 4, 578-588.
Paice, C.D. & Jones, P.A. (1993). The identification of important concepts in highly structured technical papers. ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'93, 69-78.
Papineni, K., Roukos, S., Ward, T. & Zhu W.J. (2001). Bleu: a method for automatic evaluation of machine translation. IBM Research Division, Thomas J. Watson Research Centre.
Robertson, S.E. (1990). On term selection for query expansion. Journal of Documentation, 46, 4, 359-364.
Somlo, G. L. & Howe, A. E. (2003). Using web helper agent profiles in query generation. International Conference on Autonomous Agents and Multiagent Systems, AAMAS'03, July, Melbourne, Australia, 812-818.
Spink, A. & Saracevic, T. (1997). Interactive information retrieval: Sources and effectiveness of search terms during mediated online searching. Journal of the American Society for Information Science, 48(8), 741-761.
Tombros, A. & Sanderson, M. (1998). Advantages of query biased summaries in information retrieval. ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'98, 2-10.
White, R.W., Ruthven, I. & Jose, J.M. (2003). A task-oriented study on the influencing effects of query-biased summarisation in web searching. Proceedings of Information Processing and Management, 707-733.
Williams, H.E., Zobel, J. & Bahle, D. (2004). Fast phrase querying with combined indexes. ACM Transactions on Information Systems, 22(4), 573-594.
Yeh, J.Y., Ke, H.R., Yang, W.P. & Meng, I.H. (2005). Text summarisation using a trainable summariser and latent semantic analysis. Information Processing and Managements, 41, 75-95.