Pseudo-Relevance Feedback and Statistical Query Expansion for Web Snippet Generation

Youngjoong Ko¹, Hongkuk An², and Jungyun Seo²

¹Dept. of Computer Engineering, Dong-A University,
840 Hadan 2-dong, Saha-gu,
Busan, 604-714, Korea
yjko@dau.ac.kr,

²Dept. of Computer Science and Program of Integrated Biotechnology, Sogang University,
Sinsu-dong 1, Mapo-gu,
Seoul, 121-742, Korea
anhg99@sogang.ac.kr, seojy@sogang.ac.kr

1. Corresponding author : Youngjoong Ko

2. Corresponding address : Department of Computer Engineering, Dong-A University, 840, Hadan
2-dong, Saha-gu, Busan, 604-714, Korea

3. Corresponding telephone number : 82-51-200-7782

4. Corresponding fax number : 82-51-200-7783

5. Corresponding Email address : yjko@dau.ac.kr
**Pseudo-Relevance Feedback and Statistical Query Expansion for Web Snippet Generation**

**Abstract**

A (page or web) snippet is a document excerpt allowing a user to understand if a document is indeed relevant without accessing it. This paper proposes an effective snippet generation method. A statistical query expansion approach with pseudo-relevance feedback and text summarization techniques are applied to salient sentence extraction for good quality snippets. In the experimental results, the proposed method showed much better performance than other methods including those of commercial Web search engines such as Google and Naver.

*Keywords:* Information Retrieval; Text Summarization; Pseudo-Relevance Feedback; Query Expansion; Snippet Generation

1 **Introduction**

Today’s Web search engines use text summaries to help users make relevance decisions. Most summaries, such as those by Google, are based on the query terms from the user’s search: query-biased summaries. Since these Web search engines still require users to read many closely-ranked documents to search for their information seeking goals, the query-biased summaries would be helpful in Information Retrieval or topic-detection. Snippet summaries are query-biased summaries in Web search engines that show which query terms appear in a document and the words around those query terms [5].

This paper investigates the quality of snippets from commercial Web search engines and proposes a new effective snippet generation method. The proposed method is based on pseudo-relevance feedback.
The pseudo-relevance feedback technique is a well known approach to query expansion in Information Retrieval. This pseudo-relevance feedback technique is applied to a query-biased summarization task to create high-quality snippets. In the experiments, we constructed two data sets from two commercial search engines: Google and Naver. The proposed method showed significantly improved performance in both the data sets.

The rest of this paper is organized as follows. Section 2 presents previous related work. In section 3, we explain the proposed query-biased summarization method for snippet generation. Section 4 is devoted to the analysis of empirical results. Finally, the conclusion is described.

2 Related Work

Several previous studies were carried out for generating text summaries by using query expansion [3,7]. Sanderson’s summarizer was based on query expansion from the Local Context Analysis (LCA) of the INQUERY retrieval system, but the use of LCA expansion terms provided no improvement to the summary quality. Goldstein et al. used several variations of pseudo-relevance feedback which added the title or the top ranked sentence to the query, and they obtained significant improvements in their experiments. Their methods were compared to the proposed method in our experiments.

The statistical approaches for sentence extraction generally consider the surface level features of a sentence as important ones: title, location, length, frequency, and so on [11,12]. That is, a sentence similar to the title is a salient sentence, the leading several sentences of a document are more important, a long sentence is regarded as more important, and a sentence including high frequency terms is more important.

3 The Query-biased Summarization Using Pseudo-Relevance Feedback

While Information Retrieval finds and ranks documents based on maximizing relevance to the user’s query, query-biased summarization extracts salient sentences or sections of a document based on query-

---

1 Naver is the dominant Web search engine in Korea.
term similarity. The idea of pseudo-relevance feedback is to take the results that are initially returned from some query and to use information about whether those results are relevant to perform a new query. To apply the pseudo-relevance feedback technique to query-biased summarization, first we must find the relevant sentences automatically. In the query-biased summarization, words or phrases from sections of text are compared to the user’s query terms, which are mostly related to the user’s interest. From this point of view, a sentence including a query term can be regarded as a relevant sentence. All the noun terms of the relevant sentences become candidates for query expansion; the original query term is also included in the candidate terms. The Term Selection Value (TSV) of each candidate term is estimated by a relevance weighting function in a probability model [6]. The top \( k \) candidate terms, which contain the original query term, are selected for query expansion. The importance score of a sentence is estimated by the sum of the TSVs of all the expanded query terms in the sentence and the location information of the sentence. The location information is regarded as an important feature for estimating the importance of each sentence. It has also been used as an effective feature in the text summarization literature [11]. The salient sentences are extracted according to the calculated importance score and the snippet of each document is created from them. The whole process is summarized as follows:

1. Each sentence is segmented, and it is analyzed by a POS tagger to extract noun terms as the candidate terms of query expansion.

2. Relevant and non-relevant sentences are separated by whether each sentence includes a query term.

3. The relevance weight of candidate terms is estimated by the statistical weighting function and the initial query is expanded by using the candidate terms with the high relevance weight (TSV).

4. The importance score of each sentence is estimated by using the TSVs of expanded query terms and the location information of that sentence.

5. Finally, a snippet is generated by sentences with a high importance score.
3.1 Query Expansion Using Statistical Relevance Weighting for Summarization

All the noun terms from relevant sentences become candidate terms for query expansion. To choose salient terms among the candidate terms, the relevance weight (TSV) of each term is estimated by the following function depending on the distribution of candidate terms in relevant and non-relevant sentences. This is based on the basic probability model proposed by Robertson and Sparck Jones [4,6,8].

\[
wt = TSV_t = \log \frac{p(1-q)}{q(1-p)} = \log \frac{(r + 0.5)(S - s + 0.5)}{(R - r + 0.5)(s + 0.5)}
\]

(1)

\(w_t\): the relevance weight of candidate term \(t\)

\(p\): the probability that candidate term \(t\) is assigned within the set of relevant sentences for initial query \(Q\)

\(q\): the probability that candidate term \(t\) is assigned within the set of non-relevant sentences for initial query \(Q\)

\(R\): the number of relevant sentences for initial query \(Q\)

\(r\): the number of relevant sentences for initial query \(Q\) having term \(t\)

\(S\): the number of non-relevant sentences for initial query \(Q\)

\(s\): the number of non-relevant sentences for initial query \(Q\) having term \(t\)

After all the candidate terms are sorted according to their TSVs, the top \(k\) candidate terms are selected as expanded query terms. \(k\) is set to six by a simple experiment in this paper.

3.2 The Importance Score Estimation of Each Sentence for Generating a Snippet

All the sentences including initial query terms become candidates to be extracted as a query-biased summary for a snippet. This section describes how to extract salient sentences from the set of the candidate sentences. The importance score of each candidate sentence is estimated by using the TSVs of expanded
query terms and the location information of the sentence within a document. The final importance score is calculated by formula (3).

$$\text{RWscore}(S_i) = \sum_{eqt_j \in S_i} w_{eqt_j}$$  \hspace{1cm} (2)

$$\text{score}(S_i) = \alpha \times \frac{\text{RWscore}(S_i)}{\text{RWscoreMax}} + (1 - \alpha) \times (1 - \frac{i-1}{N})$$  \hspace{1cm} (3)

In formula (2), $w_{eqt_j}$ denotes the relevance weight (TSV) of $j$-th expanded query term ($eqt_j$) included in a candidate sentence, and $\text{RWscore}(S_i)$ denotes the importance score which is calculated by the sum of relevance weights (TSVs) of expanded query terms in the candidate sentence, $S_i$. By formula (3), the final importance score is estimated by the linear combination of a normalized $\text{RWscore}(S_i)$ and a location score. $\text{RWscoreMax}$ denotes the maximum $\text{RWscore}$ value within a document. $N$ is the total number of sentences in the document and $S_i$ denotes the $i$-th located sentence. The parameter $\alpha$ is set to 0.4 by a simple experiment. As a result, the top $m$ candidate sentences are selected as a summary for a snippet; $m$ is differently chosen for each document because it is the same number as the number of sentences of a snippet for the document from Google and Naver.

4 Evaluation

4.1 Data Sets and Performance Measures

Two data sets consisted of news articles and their snippets from Google (www.google.co.kr) and Naver (www.naver.com) search engines. Four categories (science & technology, sports & entertainment, politics & economy, and society & culture) were selected. Twenty queries were extracted from the popular query list of Naver for each category and they were applied to Google in the same manner. Some representative
queries are as follows: ‘RFID’ for science & technology, ‘Ji-sung Park (Korean soccer player of Manchester United)’ for sports & entertainment, ‘buk-haek (North Korea’s nuclear weapon)’ for politics & economy, and ‘dock-gam (Influenza)’ for society & culture’. At first, the top 10 Web documents from the result of each search engine were chosen as test data; some documents can occur in both data sets concurrently because we used the same query set for each search engine, but most of them have different snippets. The data set consists of 80 queries and 1,600 documents. Some of these 1,600 documents were removed because they have the same number of sentences including query terms as the snippet of each search engine. That is, only sentences including a query term are qualified as a snippet, and the number of selected sentences for each document is equal to the number of sentences from its Google or Naver snippet in the experiments. Thus, if a document has only a few sentences with query terms, its accuracy must always be 100% regardless of selecting any sentence and applying any summarization technique.

The final data sets are composed of 80 queries and 1,126 documents: Google (566) and Naver (560). A validation set was used to set up several parameters ($k$: the number of expanded query terms in section 3.1 and $\alpha$: the parameter of formula (3)) of the proposed method. It consists of 30% of the documents which are randomly chosen from our data set. Three people participated in extracting the most salient sentence from a document by a voting mechanism for evaluation; a document and its related query are provided to them for this process. Table 1 shows the distribution of each category in the data sets.

<table>
<thead>
<tr>
<th>Category</th>
<th>Validation Set</th>
<th>Test Data</th>
<th>Total Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>science &amp; technology</td>
<td>77</td>
<td>181</td>
<td>258</td>
</tr>
<tr>
<td>sports &amp; entertainment</td>
<td>106</td>
<td>248</td>
<td>354</td>
</tr>
<tr>
<td>politics &amp; economy</td>
<td>65</td>
<td>153</td>
<td>218</td>
</tr>
<tr>
<td>society &amp; culture</td>
<td>88</td>
<td>208</td>
<td>296</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>336</strong></td>
<td><strong>790</strong></td>
<td><strong>1,126</strong></td>
</tr>
</tbody>
</table>

Table 1. The category distribution of the data set
As summaries from the proposed method are compared to those from Google or Naver, the number of sentences in a summary extracted by the proposed method is equal to that in a snippet from Google or Naver for fair comparison. In addition, the actual shape of a snippet from Google or Naver is composed not of complete sentences but as the part of a sentence including the query term. Thus we considered the appearance of a part of sentence as that of a complete sentence for fair evaluation. The accuracy is used as a performance measure in this paper. If an extracted summary or a snippet from Google or Naver includes the most salient sentence selected by a person, it is regarded as a correct summary. Otherwise, it is regarded as an incorrect summary.

4.2 Other Query-biased Summarization Approaches Compared to the Proposed Method

The proposed query expansion method using pseudo-relevance feedback is verified by comparison with other query expansion methods using pseudo-relevance feedback: the title method and the top ranked sentence method. The former adds the title to the initial query to create an expanded query and the latter adds the top ranked sentence to the initial query [3]. The inner product is used as a similarity measure to choose the top ranked sentence. The importance score of each sentence is measured by the similarity between the expanded query and each sentence as follows:

\[
Sim(EQ, S_i) = \sum_{j=1}^{n} bw_{eqj} \cdot bw_{t_j}
\]  

(4)

where \(bw_{eqj}\) denotes the binary weight (presence or absence) of \(j\)-th expanded query term (\(eqt_j\)) in the expanded query vector with the title or top ranked sentence, \(EQ\), and \(bw_{t_j}\) denotes that of \(j\)-th term (\(t_j\)) in a sentence vector, \(S_i\).
The final title method and the final top ranked sentence method exploit a similar formula to formula (3) except using $Sim(EQ,S_i)$ instead of $RWscore(S_i)$ as follows:

$$score(S_i) = \alpha \times \frac{Sim(EQ,S_i)}{SimMax} + (1 - \alpha) \times (1 - \frac{i - 1}{N})$$  \hspace{1cm} (5)

Another query-biased summarization method is compared to the proposed method in this paper. It is based on sentence extraction using TF-ISF (Term Frequency-Inverted Sentence Frequency), which is a variation of the well known TF-IDF (Term Frequency-Inverted Document Frequency) in Information Retrieval. All the noun terms of the candidate sentences, which include a query term, are weighted in the following formula (6).

$$w_i = tf_i \times (\log(N/n_i) + 1)$$  \hspace{1cm} (6)

where $tf_i$ denotes the term frequency of term $t$, $N$ denotes the number of sentences in a document, and $n_i$ denotes the number of sentence including term $t$.

The final importance score of each candidate sentence is calculated by formula (8).

$$TFISFscore(S_i) = \sum_{w_j \in S_i} w_j$$  \hspace{1cm} (7)

$$score(S_i) = \alpha \times \frac{TFISFscore(S_i)}{TFISFscoreMax} + (1 - \alpha) \times (1 - \frac{i - 1}{N})$$  \hspace{1cm} (8)

In formula (7), $TFISFscore(S_i)$ means the importance score which is calculated by the sum of the TF-ISF values of all the noun terms in a sentence, $S_i$. 
4.3 Experimental Results

4.3.1 The Experiments for Setting Parameters

The purpose of the experiments in this section is to make a decision about setting several parameters ($k$ and $\alpha$) before the main experiments are conducted. As described in section 4.1, a validation set is used in these experiments and the importance score of each candidate sentence is estimated by only the normalized sum of TSVs ($RWscore$). Firstly, we observed the performance changes of the proposed method according to the number of the expanded query terms ($k$: 1 to 7). As you can see in Table 2, the best performance is achieved when 6 expanded query terms are used and stable performances are observed from using over 4 expanded query terms. This experiment result gives us proof of the effectiveness of query expansion in summarization; one expanded query term usage (46.1%) means that only the original query term was used in the experiment. As a result, we achieved 7.8% improvement when we used 5 more query terms by query expansion.

![Table 2. The performance changes according to the number of expanded query terms in the validation set](image)

<table>
<thead>
<tr>
<th>The number of expanded query terms</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>46.1%</td>
<td>46.1%</td>
<td>49.5%</td>
<td>52.5%</td>
<td>52.6%</td>
<td>53.9%</td>
<td>52.6%</td>
</tr>
</tbody>
</table>

After $k$ was set to six, we carried out an additional experiment to set up the $\alpha$ value for adding location information into the final importance score calculation. As a result, we obtained the best performance (68.75%) when the 0.4 value is used as $\alpha$. This accuracy value is an improvement as high as 14.85% over the proposed system without location information. This is the obvious proof that location information is effective in snippet generation.
4.3.2 Experimental Results to Verifying the Proposed Method in Each Data Set

Tables 3 and 4 show that the proposed method outperforms Google and Naver in accuracy. Moreover, the proposed query expansion method is also much superior to other query-biased summarization methods in both the data sets. In Tables 3 and 4, the performance results are reported as accuracy, and each number in parentheses denotes the number of corrected documents in total test documents.

**Table 3. Results in the Google data set**

<table>
<thead>
<tr>
<th>Title method</th>
<th>Top ranked sentence method</th>
<th>TF-ISF method</th>
<th>Proposed method</th>
<th>Search engine</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57.43% (228)</td>
<td>55.16% (219)</td>
<td>60.7% (241)</td>
<td><strong>68.26% (271)</strong></td>
<td>56.17% (223)</td>
<td>397</td>
</tr>
<tr>
<td>Total Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58.1% (329)</td>
<td>55.8% (316)</td>
<td>61.3% (347)</td>
<td><strong>68.7% (389)</strong></td>
<td>59.5% (337)</td>
<td>566</td>
</tr>
</tbody>
</table>

**Table 4. Results in the Naver data set**

<table>
<thead>
<tr>
<th>Title method</th>
<th>Top ranked sentence method</th>
<th>TF-ISF method</th>
<th>Proposed method</th>
<th>Search engine</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56.48% (222)</td>
<td>57.5% (226)</td>
<td>58.52% (230)</td>
<td><strong>66.92% (263)</strong></td>
<td>19.84% (78)</td>
<td>393</td>
</tr>
<tr>
<td>Total Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57.1% (320)</td>
<td>56.8% (318)</td>
<td>60.9% (341)</td>
<td><strong>67.1% (376)</strong></td>
<td>20.4% (114)</td>
<td>560</td>
</tr>
</tbody>
</table>

In spite of the outstanding accuracy of the proposed method, it has one weakness: more execution time. The request of more execution time can make it difficult to apply the proposed method to practical Information Retrieval systems. Execution efficiency has always been a major concern of Information Retrieval systems since most of them are interactive, and a long retrieval time will interfere with the usefulness of the system [2]. However, the developing trend of recent search engines gives us insight into this problem. That is, since most search engines analyze input queries to provide the order information of popular queries to users or for other reasons, the snippets of at least popular queries can be previously made and provided to users by using the proposed method in batch process. These kind of query cache methods have been studied to store pre-computed final result pages for very popular queries and fetch
straight from the query cache when one of the popular queries is issued [1,9]. Moreover, Turpin et al. presented algorithms and data structures for efficient generation of query-biased snippets [9]. They reduced snippet generation time by reordering/compressing documents and caching documents. We think our proposed algorithm can be applied to a part of Turpin et al.’s algorithms as well as query caching; a part of our algorithm can help improve their scoring sentence algorithm and sentence reordering algorithm.

5 Conclusion

This paper presents the outstanding snippet generation method using the pseudo-relevance feedback technique. The effectiveness of the proposed method was verified by the experiments on the Google and Naver data sets. Finally, the proposed method achieved much better performance than Google and Naver as well as other query-biased summarization methods.

Acknowledgement

This research (paper) was performed for the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Ministry of Knowledge Economy of Korea.

References


