

Automated Query-biased and Structure-preserving Text Summarization on Web Documents

F. Canan Pembe

*Dept. of Computer Engineering
Boğaziçi University, İstanbul, Turkey
canan.pembe@boun.edu.tr*

Tunga Güngör

*Dept. of Computer Engineering,
Boğaziçi University, İstanbul, Turkey
gungort@boun.edu.tr*

Abstract

Automatic summarization has become an important application recently due to the increased amount of information available on the Web. Summarization techniques can be very useful in improving the effectiveness of Web search. However, the available search engines, such as Google, only display short extracts under the search results, e.g. two lines of text fragments which consist of the query words and their surrounding text. In this paper, we investigate novel summarization techniques to improve the effectiveness of search engines. The proposed system incorporates the structure of the documents, namely the sectional hierarchy, into the output summaries. Different from the previous work, both the structural information and the content to be displayed in the summary are selected in a query-biased way. The system also uses natural language processing techniques for summarization purposes such as identification of phrases as better content carriers than single words.

1. Introduction

Automatic summarization is in fact an old area of research dating back to late 1950s. However, there has been an increasing attention to this field from government, academia and industry in recent years. The reason is the rapid growth of accessible information sources, mostly the World Wide Web, resulting in a well-known problem of information overload [1]. This means that the available information needs to be efficiently used and there is no time to read everything individually. At this point, the process of automatic summarization gains importance. However, the idea of creating summaries as successful as humans may be a long-term research direction. In parallel, summaries which are not as perfect as human summaries can be utilized in improving the effectiveness of other tasks [2]. One such task is

information retrieval; that is, the task of finding relevant documents in a large collection in order to satisfy user requests (queries) for particular items of information.

A typical search engine such as Google may return a huge number of results, i.e. links to Web pages, in response to a user query where each result is presented together with a short summary of its contents [3]. For example, consider the output of the search engine in response to the query “artificial intelligence applications”. A user needs to scroll down the huge number of results, have a look at the short summaries (usually two lines of text fragments) under the titles one by one and click only those that seem to be relevant to his or her real information need. The summaries can be very useful in determining the relevance of each result with respect to the query. However, in practice, such short summaries are usually inadequate for the user to determine the relevance of the documents. Therefore, mostly, the user has to open a particular link to learn about the actual relevance of that document. This has several disadvantages. First, loading the page takes time. Second, if the target page is long and complex, the determination of the relevancy may be both difficult and time consuming. Also, there are usually a big number of returned results and it is not feasible for the user to open each particular link to find out its relevance. To overcome these problems and improve the effectiveness of Web search, better approaches regarding the summarization are needed. In this paper, we present a novel approach to summarization of Web documents using both the content and the structure of documents in a query-biased way.

The rest of the paper is organized as follows. First, a brief overview of related work is given. This is followed by the description of the proposed system together with the structural and linguistic processing of documents. Then, the current implementation of the system is presented. This is followed by the conclusion and future work.

2. Related work

Currently available major Web search engines use short summaries of document contents in displaying their results [3, 4]. Google creates document summaries using query-biased techniques. Query words appearing within the document are output together with some of their context; i.e., with leading and trailing non-query terms, concatenated with "...".

WebDocSum is a retrieval interface providing longer query-biased summaries to improve the search experience of Web users [5]. The system uses surface-level extraction techniques; that is, it scores and selects sentences based on features such as title, location, relation to query and text formatting for the output summary. A novelty of the system is the longer summaries it provides as an adjunct to search engines by means of a summary window. Instead of two lines of summaries displayed under the search results, only the link titles are listed and the corresponding summaries are presented in a separate summary window when the mouse is moved on a particular link. The summarization system is shown to be more effective than the summaries of Google and Altavista based on a task-based evaluation. WebDocSum uses a query-biased technique for summarization. However, different from our work, it does not incorporate the structure of documents into the output summaries.

Another related work uses also the structure of documents in the output summaries [6]. The system builds a "table of content"-like hierarchy of sections and subsections for each document using some heuristics on HTML tags present in the documents and incorporates this structural information in the output summaries. However, that system only creates general-purpose summaries, not tailored for particular user queries or Web search task.

3. Proposed system

In order to improve the effectiveness of Web search, the following system regarding the summarization is proposed. First, the length of the summaries provided in search results is increased. However, if these longer summaries were again displayed under the corresponding titles, then the user would need to scroll too much to see consecutive results. To prevent this problem, similar to a previous work in the literature [5], only the titles of each link are listed, and in a separate frame, the summary for that document is displayed when the user moves the mouse on a particular link. Although the summaries in this approach are much longer than the traditional approaches, they are still limited; e.g., to the size

of the area of the screen that can be seen without scrolling.

We also take into account the fact that Web documents are not just flat texts, but rather they usually bear a structure; i.e. they consist of some sections and subsections; e.g. the abstract and introduction sections (See Figure 1 for an example). Therefore, in the proposed system, some of this structural information is incorporated into the summaries in order to help the users to judge the relevancy of each document more effectively. A novelty of the proposed system is the use of this structural information for providing the context of the text fragments, which are selected as a part of the summary, in a query-biased way, as will be detailed later in the text.

The proposed system also exploits natural language processing techniques for summarization purposes as well as the traditional term-frequency statistics. In the following subsections, the structural processing, the linguistic processing and the summarization algorithm used by the proposed system will be given.

Automated Query-biased and Structure-preserving Text Summarization on Web Documents

Context keywords:
Automatic Summarization, Natural Language Processing,
Human-Computer Interaction

Abstract

Automatic summarization has become an important application recently due to the increased amount of information available on the Web. Summarization techniques can be very useful in improving the effectiveness of Web search. However, the available search engines, such as Google, only show a limited capability in summarizing the Web documents, e.g. displaying only two lines of text fragments which contain the query words and their surrounding text as the summary. In this paper, we investigate novel summarization techniques to improve the effectiveness of search engines. The proposed system incorporates the structure of the documents, namely the sectional hierarchy, into the output summaries. Different from the previous work both the structural information and the contents to be displayed in the summary are selected in a query-biased way. The system also uses natural language processing techniques for summarization purposes such as identification of phrases as better context carriers than single words.

1. Introduction

Automatic summarization is in fact an old area of research dating back to late 1950s. However, there has been an increasing attention to this field from government, academia and industry in recent years. The reason is the rapid growth of accessible information sources, namely the World Wide Web, resulting in a well-known problem of information overload [1]. This means that the available information needs to be efficiently used and there is no time to read everything individually. At this point, the process of automatic summarization gains importance. However, the idea of creating summaries as successful as humans may be a long-term research direction. In parallel, summaries which are not as perfect as human summaries can be utilized in improving the effectiveness of other tasks [2]. One such task is information retrieval, that is, the task of finding relevant documents in a large collection in order to satisfy user requests (queries) for particular items of information.

A typical search engine such as Google may return a huge number of results, i.e. links to Web pages, in response to a user query where each result is presented together with a short summary of its contents [3]. For example, consider the output of the search engine in response to the query "artificial intelligence applications". A user needs to scroll down the huge number of results, have a look at the short summaries (usually two lines of text fragments) under the titles one by one and click only those that seem to be relevant to his or her real information need. The summary can be very useful in determining the relevance of each result with respect to the query. However, in practice, such short summaries are usually inadequate for the user to determine the relevance of the documents. Therefore, usually, the user has to open a particular link to learn about the actual content of that document. This has several disadvantages. First, loading the page takes time. Second, if the target page is long and complex, the determination of the relevancy may be both difficult and time consuming. Also, there are usually a big number of result titles and it is not feasible for the user to open each particular link to find out its relevance. To overcome these problems and improve the effectiveness of Web search, better approaches regarding the summarization are needed. In this paper, we present a novel approach to summarization of Web documents using both the content and the structure of documents in a query-biased way.

The rest of the paper is organized as follows. First, a brief overview of related work is given. This is followed by the description of the proposed system together with the structural and linguistic processing of documents. Then, the current implementation of the system is presented. This is followed by the conclusion and future work.

2. Related work

Currently available major Web search engines use short summaries of document contents in displaying their results (e.g. 3,4). Google creates document summaries using query-biased techniques. Query words appearing

Figure 1. An example structured document

3.1. Structural processing

Currently, most of the documents on the Web are formatted in HTML. Usually, the documents are not prepared as flat text with all the content in a fixed format. Instead, they are designed as consisting of sections and subsections using a

limited number of HTML formatting tags. Some of these tags include:

- Bold ()
- Underlined (<u>)
- Font (): together with the size attribute to specify the size of the font used
- Heading: <h1>, <h2>, <h3>, <h4>, <h5> and <h6> for different levels of headings

The structure of a document may be considered as a hierarchy, where each document has sections; each section has subsections, and so on. In the proposed system, the sections and subsections are identified using some heuristics on HTML tags. When a document contains heading tags, it is easier to identify the headings and corresponding subsections. The heading tags <h1> through <h6> usually correspond to headings of different levels in the hierarchy; e.g. <h1> is a first level heading and <h2> is a second level heading. In some documents, section headings are displayed using bold and relatively larger fonts. Section headings can also be identified using these relative features.

The algorithm to automatically find the structural properties of a document in the proposed system is as follows. First, each document is processed to identify potential headings based on HTML annotations of the document. A demonstration of the algorithm is shown on an example in Table 1. In the table, the starting and ending positions of potential headings and their HTML formatting for certain tags are stored; e.g. h1 for heading of level 1, b for bold, etc. (the features are marked as 1 for existence of that tag between the specified offsets and 0 for the inexistence). The next step is to identify different levels of headings based on heuristics on the position of occurrence within the document and formatting of the potential headings. For example, the headings in the same level will probably have the same formatting according to the specified features. Based on these findings, the headings used within the text can also be numbered hierarchically, e.g. the first level headings, 1, 2, 3, the second level headings as 1.1, 1.2, 2.1, etc. as seen in Table 1. Once the different levels of headings within the document are identified, the sections and subsections of the document can also be identified accordingly. For instance, a section starts with its heading and ends with the consecutive heading in the same level.

In structural processing step, each document is annotated with XML tags according to its structure, such as its sections and subsections. Also, each sentence is identified and tagged. An example document annotated with structural tags is given in Figure 2.

Table 1: Analysis of potential headings in a document

start	end	h1	H2	h3	h4	h5	h6	b	a	u	f	level	number.
0	32	1	0	0	0	0	0	0	0	0	5	1	1
89	106	0	0	0	0	0	0	1	0	1	3	2	1.1
200	228	0	0	0	0	0	0	1	0	0	3	3	1.1.1
410	418	0	0	0	0	0	0	1	0	0	3	3	1.1.2
419	430	0	0	0	0	0	0	0	1	0	3	4	1.1.2.1
490	503	0	0	0	0	0	0	0	1	0	3	4	1.1.2.2
670	689	0	0	0	0	0	0	1	0	0	3	3	1.1.3
740	801	0	0	0	0	0	0	1	0	1	3	2	1.2
902	930	0	0	0	0	0	0	1	0	0	3	3	1.2.1
...

```

<document>
  <heading> Automated Query-biased and Structure-
preserving Text Summarization on Web Documents
</heading>
  ...
  <section level = 1>
    <heading>Proposed System</heading>
    ...
    <section level = 2>
      <heading>Structural Processing</heading>
      ...
      <sentence>The structure of a document
may be considered as a hierarchy, where each
document has sections; each section has
subsections, and so on.</sentence>
      ...
    </section>
  ...
</section>
</document>

```

Figure 2. XML representation of a document tagged according to its structure

3.2. Linguistic processing

In the proposed system, both documents and queries are processed linguistically. First, each word is processed morphologically. Then, part-of-speech tagging is performed for each sentence. For example, consider the text portion *Automatic text summarization is an old area of research* annotated with part-of-speech tags as in the following:

Automatic/adj text/noun summarization/noun
is/verb an/det old/adj area/noun of/prep
research/noun...

Then, noun phrases, which are important content carriers in almost every natural language, can be identified using finite-state transduction based on predefined patterns. For example, an English noun phrase can be defined as follows where * is used for multiple occurrences of the components [7]:

$$NP = \text{det}^* \text{pre}^* \text{head} \text{post}^*$$

In this representation, the components of the noun phrase are:

- det (determiner): e.g. article, number
- pre (premodifier): e.g. adjective, noun
- head: usually a noun
- post (postmodifier): e.g. prepositional phrase

Such patterns based on part-of-speech tags are used to identify the phrases; e.g., the noun phrase *automatic text summarization* is composed of the components *automatic* (premodifier adjective), *text* (premodifier noun) and *summarization* (head noun).

The use of noun phrases for summarization purposes promises an increase in the effectiveness. However, the problem of syntactical variation should be considered. For example, all the phrases *text summarization*, *automatic summarization*, *automatic text summarization* are different syntactical formulations with similar meaning. Therefore, some form of regularization is necessary for phrases; i.e., syntactical normalization [7]. A normalization technique that can be used for this purpose is the extraction of word pairs with head-modifier relation from noun phrases [8]. For example, the phrase *automatic text summarization* can be normalized to the head-modifier pairs “*automatic* (modifier) + *summarization* (head)” and “*text* (modifier) + *summarization* (head)”, which can be easily found using heuristics. Such subcompounds can be used in scoring sentences in the summarization stage as better content carriers than just single words. All the information obtained in linguistic processing stage is also stored in the XML representation for later processing..

3.3. Summarization engine

In the proposed system, retrieved documents are summarized according to the user query. As a simple approach, the sentences in a given document are scored based on the frequency counts of terms (words or phrases). The sentences containing the query phrases are given higher scores than the ones containing single query words. Then, the sentences with highest scores are incorporated into the output summary together with their structural context.

Portions of text may be extracted from different sections or subsections. The resulting summary is the union of such extracts. The number of extracted sentences and the extent to which their context is displayed depends on the summary frame size which is fixed to the size of the screen that can be seen without scrolling, such as 35 lines of text. In the sentence extraction algorithm, whenever a sentence is selected for the inclusion in the summary, some of the headings in that

context are also selected. The algorithm is as follows:

Algorithm:

- 1: Rank all the sentences according to their score.
 - 2: Add the main title of the document to the summary.
 - 3: Add the first level-1 heading to the summary.
 - 4: **While** (summary size limit not exceeded)
 - 5: Add the next highest scored sentence.
 - 6: Add the structural context of the sentence:
 (if any and not already included in the summary)
 - 7: Add the highest level heading above the
 extracted text (call this heading *h*).
 - 8: Add the heading before *h* in the same level.
 - 9: Add the heading after *h* in the same level.
 - 10: Repeat steps 7, 8 and 9 for the next highest level
 headings.
 - 11: **End while**
-

4. Implementation

We used GATE framework for text engineering as the underlying development environment which is an open source project using component-based technology in Java [9,10]. GATE is being used as an infrastructure by many academic and commercial projects. Using such a framework has several advantages because it includes commonly used natural language functionalities such as part-of-speech tagging and it is a modular environment into which new components can be easily added.

In the implementation, the Google API provided as a GATE plug-in is used to query Google and build the document corpus that contains the search results returned by Google for the query. Then, ANNIE English Tokeniser, Sentence Splitter, and Part-of-Speech Tagger provided as a part of GATE are used to split the text into individual tokens, sentences and to produce a part-of-speech tag on each token, respectively. GATE also has built-in finite-state transduction capabilities. The transducer runs based on grammars written in a language called JAPE (Java Annotations Pattern Engine) language. Then, to identify phrases, only patterns need to be described. In order to identify noun phrases, Noun Phrase Chunker provided as a GATE plugin is used.

We have written a summarization engine module which is able to create summaries using the linguistic and structural information as explained in the previous section. In the summaries output by the proposed system, consecutive fragments and headings are separated with dots (...) indicating that more material follows in between. A sample summary is given in Figure 3 for a demonstration of the proposed system. As can be seen, both the document structure and the highest scored fragments with

important phrases according to the user query are incorporated into the summary. In this way, it is expected that the user can judge the relevance of each document better than the traditional approaches without the necessity to load the actual page.

Automated Query-biased and Structure-preserving Text Summarization on Web Documents

1. Abstract
 ...
 Different from the previous work, both the structural information and the content to be displayed in the summary are selected in a query-biased way.
 ...

2. Related Work
 ...

3. Proposed System
 ...

3.1. Structural Processing
 ...
 The structure of a document may be considered as a hierarchy, where each document has sections; each section has subsections, and so on.
 ...

3.2 Linguistic Processing
 ...

Figure 3. An example to demonstrate the proposed system

5. Conclusion

This paper presented the design and implementation of an automated summarization system for Web search, as an attempt to improve the search experience of users. The system uses structural and linguistic information obtained from the documents both in the summarization process and in the output summaries. As a novelty, the system uses this information in a query-biased way within the Web search context.

The research can be extended in several directions. First, the scoring method of sentences can be improved with query-biased methods. Second, the system can be refined considering different types of search tasks, such as searching for a particular fact or searching for background information about a subject, etc. Also, the heuristics used in the identification of structural information and incorporation in the output summary can be improved. Finally, natural language processing techniques is always open to improvement; e.g. incorporating verb phrases besides noun phrases to the system. As a future work, the system will be evaluated on a comprehensive task-based evaluation.

6. References

- [1] I. Mani and M. T. Maybury, *Advances in Automatic Text Summarization*, MIT Press, Cambridge, 1999.
- [2] K. Sparck Jones, "Automatic Summarizing: Factors and Directions", *In Advances in Automatic Text Summarization*, 1-12, MIT Press, Cambridge, 1999.
- [3] Google, 2006. <http://www.google.com>.
- [4] AltaVista, 2006. <http://www.altavista.com>.
- [5] R. W. White, J. M. Jose and I. Ruthven, "A Task-oriented Study on the Influencing Effects of Query-biased Summarization in Web Searching", *Information Processing and Management*, 39, 5, 2003, pp. 707-733.
- [6] H. Alam, A. Kumar, M. Nakamura, A. F. R. Rahman, Y. Tarnikova and C. Wilcox, "Structured and Unstructured Document Summarization: Design of a Commercial Summarizer using Lexical Chains", *In Proceedings of Seventh International Conference on Document Analysis and Recognition*, IEEE Computer Society, 2003, pp. 1147-1150.
- [7] A. Arampatzis, T. Weide, C. Koster and P. Bommel, "Linguistically Motivated Information Retrieval", *Encyclopedia of Library and Information Science*, 69, 2000, pp. 201-222.
- [8] D. A. Evans and C. Zhai, "Noun-Phrase Analysis in Unrestricted Text for Information Retrieval", *In Proceedings of 34th Annual Meeting of the Association for Computational Linguistics*, 1996, pp. 17-24.
- [9] GATE, A General Architecture for Text Engineering. <http://gate.ac.uk/>.
- [10] D. Maynard, K. Bontcheva, H. Saggion, H. Cunningham and O. Hamza, "Using a Text Engineering Framework to Build an Extendable and Portable IE-based Summarisation System", *In Proceedings of the ACL Workshop on Text Summarisation*, 2002.