SNIPIT: Summarize Needed Information Per Interrogative Text

Christopher Wienberg
Computer Science Department
Pomona College
185 E. Sixth Street
Claremont, CA 91711
christopher.wienberg@pomona.edu

Matthew Williams
Computer Science Department
Pomona College
185 E. Sixth Street
Claremont, CA 91711
matthew.williams@pomona.edu

ABSTRACT
There is a vast quantity of information available to computers on the web, and an ever growing need to search this information efficiently. Therefore, developing a system to allow users to quickly evaluate whether a given document satisfies their information need is necessary. To solve this problem, we propose our system called SNIPIT. SNIPIT examines the content of sentences in the document as well as temporal information about each sentence, and based on a user's query, picks two sentences to best convey the content of the document and how it relates to the user's query. To evaluate our system we compared our snippets to the snippets generated by *Yahoo!* on three metrics. We found our snippets to be dissimilar from *Yahoo!*'s. However, we believe this does not reflect poorly on our system given there were many complications in evaluation.

1. INTRODUCTION
Automatic text summarization is commonly used in web search engines to provide users with a better search experience. The results of a user's query are coupled with a short snippet of text that should be a representative summary of the document returned by the search engine, hopefully providing the user with more information about whether the page will be helpful or not. The general form of text summarization aims to only summarize the document, however query-biased summarization aims to generate a summary specifically focused around the user's query. This gives the user an idea of how their query is actually addressed within the returned document.

In order to obtain the snippet, text from the document may either be extracted or abstracted. If it is extracted the summary consists of solely text from the document itself. Abstracted summaries may contain words and phrases not found in the document, which requires difficult methods to handle discourse understanding and natural language processing. In this paper, we choose to use the extraction method because it may be applied in real time and it provides more than sufficient results to help the user understand the context of the document.

Various methods to decide which sentences should be extracted and used for the snippet have been researched and implemented in different summarization systems. Our method scores each sentence according to a set of feature functions. These functions take into consideration a sentence's temporal context, its relation to the query and title, and other significant words within the document.

The rest of the paper is organized as follows. We begin with a brief discussion of related work in the field in Section 2. Next, we discuss our methods in Section 3, then our evaluation procedures and results in Section 4 and Section 5 respectively. Finally, we close with a conclusion and a discussion of possible further extensions to our project.

2. RELATED WORK
There has been significant research in the area of query-based snipppet generation. Tombros and Sanderson pioneer one of the first works in the space [1], which relies on several standard summarization techniques– the title, early paragraphs, and term occurrences–to score content from the document for its inclusion in the snippet, as well as the appearance of query terms in these potential snippets. They test their system using a human study that compares their snippets to simply the title and first few sentences of the document. Human evaluators are either given the "simple" snippets of just the first few sentences, and the titles of documents, or are given their system's snippets and the titles, and are asked in a limited amount of time to identify which documents are relevant and which are not. The participants were allowed to look at the full text of the document if desired. They looked at several measures in this study: the precision (number of documents correctly deemed relevant per total documents deemed relevant) and recall (number of documents deemed relevant per total relevant docs), the speed at which users decided whether a document was relevant or not, the number of times the user referred to the full text (which would indicate a poor summary), and the user's subjective opinion of the quality of the snippets. In all these metrics, their system performs significantly better over the "simple" snippets.

Wang et. al. [2] originally implemented many of the methods we chose to use in our system in their pursuit of creating a system which learns query-biased Web page summarization. The feature functions they utilize to determine the importance of a group of words, or sentence, from the contents of the Web page are what we modeled much of our work on. Furthermore, they proceed further to incorporate the context of the links within the Web-page to determine their snippets. This combination of features functions, which looked at both context and content proved to be quite useful and indicated that together they were more powerful than each separate. However, due to the nature of our intended corpus, their Web-page specific functions which look at links are not applicable to our system.

3. METHOD

We set out to find two sentences as our snippet. We chose to use complete sentences because a sentence as a fundamental building block of our system of communication conveys a full thought. By not cutting the sentence into smaller fragments we are preserving the full thought. In discussing possible snippet generation methods, we decided it would be best to find query-based snippets that not only reflect the query, but do not differ significantly from the underlying theme of the document. The motivation behind this decision is that we would like to indicate to the user that the document may not meet their information need. We incorporated several methods of scoring sentences based on temporal and content information.

3.1 Paragraph Location

Each sentence was scored based on its paragraph's location within the document, compared to the location of other paragraphs. The scoring is based on Zipf's law: sentences in the first paragraph get 1 point, in the second get $\frac{1}{2}$, and so on. The only break with this is the final paragraph, which is scored exactly as the second paragraph is. The motivation behind this is that earlier paragraphs are likely to contain the most pertinent information in the document, and the last paragraph is likely to summarize the document (conclusion) and again contain some of the most pertinent information.

3.2 Sentence Location

Each sentence is scored based on its location within its paragraph. This process is similar to the paragraph location scoring method: earlier sentences, and the last sentence, are more likely to convey pertinent information about the document, and therefore should be more likely to appear in a snippet. Again, the scoring is based on Zipf's law, with the first sentence of a paragraph getting 1 point, the second getting $\frac{1}{2}$, and so on. Additionally, the last sentence of a paragraph is scored identically to the second, because it usually serves as either a summary of the current paragraph or a transition into the next.

3.3 Title Score

The title of a document conveys a lot about the content of a document. Therefore, sentences similar to the title should be held in higher regard than those which do not. We score each sentence based on its similarity to the title using the following formula:

$$Score_{title}(s) = i/n \tag{1}$$

where n is the total number of words in the title, and i is the number of words that are in the title and sentence s. This scoring method is taken from [2].

3.4 Query Score

Of course, since we are developing a system that returns snippets based on a user's query, a sentence is scored based on its similarity to the given query. Therefore, sentences similar to the query should be rewarded with a higher score. Our system rewards sentences based on the following equation:

$$Score_{query}(s) = 2 \cdot n^2/q \tag{2}$$

where q is the total number of query terms, and n is the number of unique query terms contained in sentence s. This is the query-scoring method suggested in [2].

3.5 Term Significance

In any given document, certain terms are more important than others. These words are the most likely to express the main theme of the document. Therefore, sentences that contain these "significant" terms are more likely to convey useful information about the document, and should be favored when generating snippets.

The challenge, then, is to identify these significant terms. In [1, 2], a means to determine significant terms, and how to value them when scoring a sentence for inclusion in a snippet is suggested; our approach draws from theirs. We begin by removing common stopwords from the document, and stemming all other words. We then count the occurrences of each term and throw out any term with an occurrence below a threshold T defined as:

$$T = 7 + I \cdot 0.1 \cdot |L - n| \tag{3}$$

where n is the number of sentences in the document, I is defined by:

$$I = \begin{cases} 0 & \text{if } 25 \leq n \leq 40 \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

and L is defined by:

$$L = \begin{cases} 25 & \text{if } n < 25 \\ 40 & \text{if } n > 40 \\ n & \text{otherwise} \end{cases} \tag{5}$$

Taking this list of significant terms, for each sentence s, we score it based on the following formula:

$$Score_{Significance}(s) = t^2/||s|| \tag{6}$$

where t is the number of significant terms appearing in sentence s. Tombros and Sanderson [1] found that in medium sized documents (containing 25 to 40 sentences) words which occur more than seven times are likely to be significant. They also found for larger and smaller documents the threshold should be scaled accordingly by ten percent of the length of the document. This scoring method has been tested and proven successful in both [2] and the research they draw this function from.

3.6 Application of Scoring Methods

Each of these scoring methods are calculated separately and summed to score each sentence. Then the two sentences with the highest scores are selected as the snippet.

4. EVALUATION STRATEGY

In order to evaluate our system, we developed an automatic evaluation system in conjunction with a group of fellow peers. We began by creating two test corpora that were generated by a script which pulls the query results and their snippets from *Yahoo!*; we generated one set of general queries of the web and the other from a search limited to the blogspot.com domain, and returned them in a text file. A list of 60 unique queries are used, and the first 20 results were saved. For each corpus, we then find the similarity between the snippets pulled from the Internet and our own through three different metrics.

First, the word overlap between the two snippets is counted and then divided by the length of our snippet to compute the percent of our snippet known to be correct. Sentences with higher word overlaps indicate similar snippets that should convey the same meaning. Second, again the word overlap is computed and is then divided by the number of words in the *Yahoo!* snippet to calculate the percentage of words missed in our snippet. Finally, we calculate the Jaccard Coefficient to determine the similarity between the two snippets using a standard metric. For each of these calculations stop words were removed due to their lack of significance.

5. RESULTS

The results of our evaluation procedure are as follows:

Metric	General Web	Blogspot-only
Percent Correct	24.9%	25.2%
Percent Missing	81.9%	83.0%
Jaccard Coefficient	11.0	10.7
Running time (ms/document)	12.7	7.21

Lower numbers are good in the Percent Missing and running time metrics; higher numbers are preferred for the other metrics.

While these numbers indicate poor performance, we believe this process of evaluation is understating the performance of our system. There were several issues that arose in evaluation. Occasionally, the text of the snippets for documents fetched from *Yahoo!* would not be contained within the document text. Obviously, our system cannot generate a snippet that fits the *Yahoo!* snippet when it lacks the necessary material. An example of this would be *Yahoo!* generating a snippet based on some peripheral text of the document; our document-fetching script would only grab the text from the main section of the web page. Also, in general, *Yahoo!* snippets are not the gold standard by which to judge all other snippets. Legitimate snippets can be generated that are just as qualitatively good, if not better, than those generated by *Yahoo!*. It is important to remember that this evaluation metric is only measuring similarity to *Yahoo!*'s snippets. As such, any break from *Yahoo!*'s strategy is likely to result in penalization. An example of such a break would be using full sentences rather than a small sequence of words as the snippet material, as we did in our system. Despite these flaws of our evaluation metric, we could determine no better, practical quantitative measure of the quality of the snippets generated by our system.

In addition to quality of performance, it is important to note the time cost of generating these snippets. This is important because the snippet generator is part of a larger, real time system that must respond quickly. As can be seen, snippets are generated very quickly. These response times will be even lower in the current application this system is designed for, as this application uses smaller documents. These times are very reasonable, and our snippet generator may be implement into any system without worry of slowing said system down.

6. CONCLUSION

We have built a system for the generation of query-biased snippets. Our methods are grounded in intuition as well as past, proven methods. Despite poor performance in our evaluation, we maintain confidence in our system. For future work, we would consider evaluating our system qualitatively. Given time constraints and technical difficulties, we were unable to perform this evaluation before the call for papers. In such an evaluation, we would ask users to rate the usefulness of our snippets in determining if a document satisfies their information need. We firmly believe such an evaluation will vindicate our work.

7. REFERENCES

[1] A. Tombros and M. Sanderson. Advantages of query biased summaries in information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 2–10, New York, NY, USA, 1998. ACM.

[2] C. Wang, F. Jing, L. Zhang, and H.-J. Zhang. Learning query-biased web page summarization. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 555–562, New York, NY, USA, 2007. ACM.