Automatic summarization of search engine hit lists

Dragomir R. Radev
School of Information, University of Michigan
550 E. University St.
Ann Arbor, MI 48109
radev@umich.edu

Weiguo Fan
University of Michigan Business School
701 Tappan St.
Ann Arbor, MI 48109
wfan@umich.edu

Abstract

We present our work on open-domain multi-document summarization in the framework of Web search. Our system, SNS (pronounced "essence"), retrieves documents related to an unrestricted user query and summarizes a subset of them as selected by the user. We present a task-based extrinsic evaluation of the quality of the produced multi-document summaries. The evaluation results show that summarization quality is relatively high and does help improve the reading speed and judge the relevance of the retrieved URLs.

1 Introduction

Online information is increasingly available at an exponential rate. According to a recent study by NetSizer (2000), the number of web hosts has increased from 30 million in Jan.1998 to 44 million in Jan. 1999, and to more than 70 million in Jan. 2000. More than 2 million new hosts were added to the Internet in Feb. 2000, according to this report. Similar Internet growth results were reported by Internet Domain Service (IDS, 2000). The number of web pages on the Internet was 320 million pages in Dec. 1997 as reported by Lawrence et al. (1997), 800 million in Feb. 1999 (Lawrence et al. 1999), and more than 1,720 million in March, 2000 (Censorware, 2000). The number of pages available on the Internet almost doubles every year.

To help alleviate the information overload problem and help users find the information they need, many search engines emerge. They build a huge centralized database to index a portion of the Internet: ranging from 10 million to more than 300 million of web pages. Search engines do help reduce the information overload problem by allowing a user to do a centralized search, but they also bring up another problem for the user: too many web pages are returned for a single query. To find out which documents are useful, the user often have to sift through hundreds of pages to find out that only a few of them are relevant. Moreover, browsing through the long list of retrieval results is so tedious that few users would be willing to go through. That's why research results have shown that search engine users often give up their search in the first try, examining no more than 10 documents (Jansen et al. 2000). It would be very helpful if an effective search engine could be designed to help classify the retrieved web pages into clusters and provide more contextual and summary information to help these users explore the retrieval set more efficiently.

Recent advances in information retrieval, natural language processing, computational linguistics make it easier to build a helpful search engine based on summaries of hit lists. We describe in this paper a prototype system, SNS, which blends the traditional information retrieval technology with the advanced document clustering and multi-document summarization technology in an integrated framework. The following steps are performed for a given query:

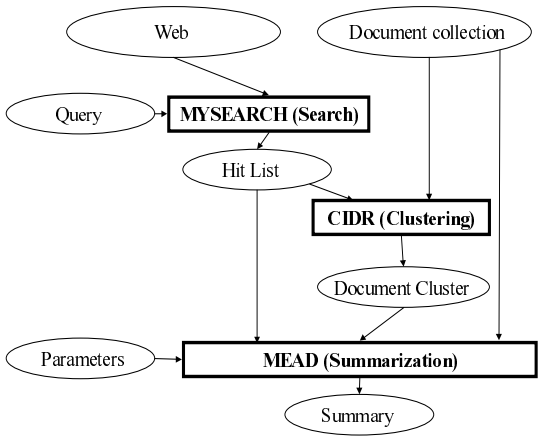

Figure 1: Architecture diagram

The general architecture of our system is shown in Figure 1. User interaction with SNS can be done in three different modes:

- Web search mode. The user enters a general-domain query in the search engine (MySearch). The result is a set of related documents (the hit-list). The user then selects which of the hits should be summarized. MEAD, the summarization component produces a cross-document summary of the documents selected by the user from the hit list.
- Intranet mode. The user indicates what collection of documents needs to be summarized. These documents are not necessarily extracted from the Web.
- Clustering mode. The user indicates that either the hit list of the search engine or a stand-alone document collection needs to be clustered. CIDR, the clustering component, creates clusters of documents. For each cluster, MEAD produces a cross-document summary.

Our paper is organized as follows. Sections 2 — 4 describe the system. More specifically: Section 2 explains how the search engine operates, Section 3 deals with the clustering module while Section 4 presents the multi-document summarizer. Section 5 describes the user interface of the system. In Section 6, we present some experimental results. After we compare our work to related research in Section 7, we conclude the paper in Section 8.

2 Search

The search component of SNS is a personalized search engine called MySearch. MySearch utilizes a centralized relational database to store all the URL indexes and other related URL information. Spiders are used to fetch URLs from the Internet. After a URL is downloaded, the following steps are applied to index the URL:

- Parse the HTML file, remove all those tags
- Apply Porter's stemming algorithms to each keyword.
- Remove stop words
- Index each keyword into the database along with its frequency and position information.

The contents of URLs are indexed based on the locations of the keywords: Anchor, Title, and Body. This allows weighted retrieval based on different word positions. For example, a user can specify that he'd like to give a weight 5 for the keyword appearing in the title, 4 for anchor, and 2 for body. This information can be saved in his personal profile and used for later weighted ranking.

Besides the weighted search, MySearch also supports Boolean search and Vector Space search (Salton, 1989). For the vector space model, the famous TF-IDF is used for ranking purpose. We used a modified version of TF-IDF: $\log(tf+0.5)*\log(N/df)$, where *tf* means the number of times a term appeared in the content of an URL, *N* is the total number of documents in the text collection, and *df* stands for the number of unique URLs in which a term appears in the entire collection.

A user can choose which search method he wants to use. He/she can also combine Boolean search with Vector Space search. These options are provided to give users more flexibility to control the retrieval results as

past research indicated that different ranking functions give different performances (Salton, 1989).

A sample search for "Clinton" using the TF-IDF Vector Space search is shown in Figure 3. The keyword "Clinton" is highlighted using a different color to help users get more contextual information. The retrieval status value is shown in a bold black font after the URL title.

3 Clustering

Our system uses two types of clustered input – either the set of hits that the user has selected or the output of our own clustering engine – CIDR (Columbia Intelligent Document Relater). CIDR is described in (Radev et al., 1999). It uses an iterative algorithm that creates as a side product so-called "document centroids". The centroids contain the most highly relevant words to the entire cluster (not to the user query). We use these words to find the most salient "themes" in the cluster of documents.

3.1 Finding themes within clusters

One of the underlying assumptions behind SNS is that when a user selects a set of hits after reading the single-document summaries from the hit list retrieved by the system, he or she performs a cognitive activity whereby he or she selects documents which appear to be related to one or more common themes. The multi-document summarization algorithm attempts to identify these themes and to identify the most salient passages from the selected documents using a pseudo-document called the cluster centroid which is computed automatically from the entire list of hits selected by the user.

3.2 Computing centroids

Figure 2 describes a sample of a cluster centroid. The TF column indicates the average term frequency of a given term within the cluster. E.g., a TF value of 13.33 for three documents indicates that the term "deny" appears 40 times in the three documents. The IDF values are computed from a mixture of 200 MB of news and web-based documents.

Term	TF	IDF	Score
app	20.67	8.90	183.88
lewinsky	34.67	5.25	182.03
currie	15.33	7.60	116.50
ms	32.00	3.06	97.97
january	25.33	3.30	83.60
jordan	18.67	4.06	75.81
referral	9.00	7.43	66.88
magaziner	6.67	10.00	66.64
Deny	13.33	4.92	65.61
Admit	13.00	4.92	63.97
monica	14.67	4.29	62.85
oic	5.67	10.00	56.64
betty	8.00	6.01	48.06
vernon	8.67	5.49	47.54
do	32.67	1.40	45.80
Telephoned	6.67	6.86	45.74
you	36.33	1.19	43.30
i	42.67	0.96	40.84
clinton	16.33	2.23	36.39
jones	11.33	3.17	35.88
or	32.33	1.09	35.20
gif	3.33	9.30	31.01
white	12.00	2.50	30.01
tripp	4.67	6.23	29.10
ctv	3.00	9.30	27.91
december	7.33	3.71	27.19

Figure 2: A sample cluster centroid

4 Centroid-based summarization

The main technique that we use for summarization is sentence extraction. We score individually each sentence within a cluster and output these that score the highest. A more detailed description of the summarizer can be found in (Radev et al., 2000).

The input to the summarization component is a cluster of documents. These documents can be either the result of a user query or the output of CIDR.

The summarizer takes as input a cluster of *d* documents with a total of *n* sentences as well as a compression ratio parameter *r* which indicates how much of the original cluster to preserve.

The output consists of a sequence of [n * r] sentences from the original documents in the same order as the input documents. The highest-ranking sentences are included according to the scoring formula below:

$$S_i = w_c C_i + w_p P_i + w_f F_i$$

In the formula, w_c, w_p, w_f are weights. C_i is the centroid score of the sentence, P_i is the positional score of the sentence, and F_i is the score of the sentence according to the overlap with the first sentence of the document.

4.1 Centroid value

The centroid value C_i for sentence S_i is computed as the sum of the centroid values C_w of all words in the sentence. For example, the sentence "President Clinton met with Vernon Jordon in January" gets a score of 243.34 which is the sum of the individual centroid values of the words (clinton = 36.39; vernon = 47.54; jordan = 75.81; january = 83.60).

$$C_i = \sum_w c_w$$

4.2 Positional value

The positional value is computed as follows: the first sentence in a document gets the same score C_{max} as the highest-ranking sentence in the document according to the centroid value. The score for all sentences within a document is computed according to the following formula:

$$P_i = \frac{(n-i+1)}{n} * \max_i(C_i)$$

For example, if the sentence described above appears as the third sentence out of 30 in a document and the largest centroid value of any sentence in the given document is 917.31, the positional value P_3 will be = 28/30 * 917.31

4.3 First-sentence overlap

The overlap value is computed as the inner product of the sentence vectors for the current sentence i and the first sentence of the document. The sentence vectors are the n-dimensional representations of the words in each sentence whereby the value at position i of a sentence vector indicates the number of occurrences of that word in the sentence.

$$F_i = \vec{S_1}\vec{Si}$$

4.4 Combining the three parameters

As indicated in (Radev & al., 2000) we have experimented with several weighting schemes for the three parameters (centroid, position, and first-sentence overlap). Until this moment, we have not come to the point in which the three weights w_c, w_p, and w_f are either automatically learned or derived from a user profile. Instead, we have experimented with various sets of empirically determined values for the weights. In this paper the results are based on equal weights for the three parameters $w_c = w_p = w_f = 1$.

5 User Interface

We describe in this section the user interface for web search mode as described earlier in Section 1.

One component of our system is the search engine (MySearch). The detailed design of the search component is discussed in Section 2. The result of a sample query "Clinton" to our search engine is shown starting in Figure 4.

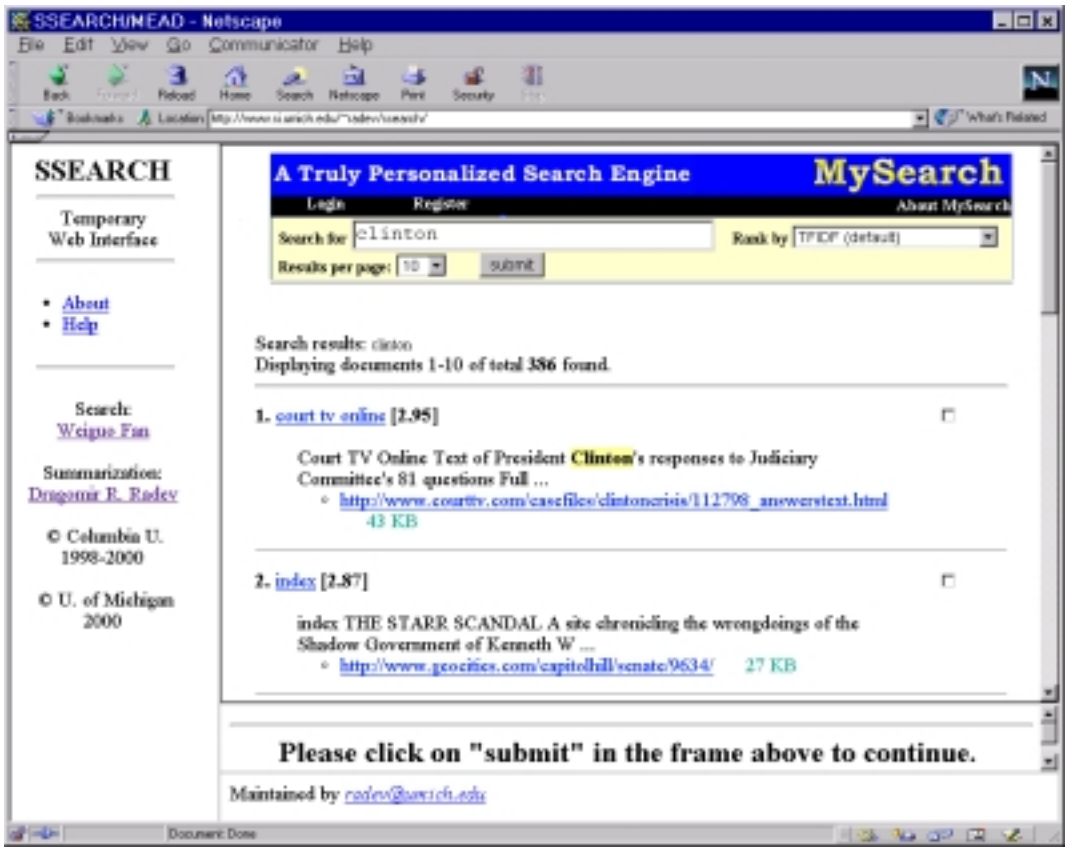

Figure 3: Sample user query

A user has the option to choose a specific ranking function as well as the number of retrieval results to be shown in a single screen. The keyword contained in the query string will be automatically highlighted in the search results to provide contextual information for the user.

The overall interface for SNS is shown in Figure 4. On the top right of the frame is the MySearch search engine. When a user submits a query, the screen in Figure 5 appears. As can be seen from Figure 5, there is a check box along with each retrieved record. This allows the user to tell the summarization engine which documents he/she wants to summarize. After the user clicks the summarization button, the summarization option screen is displayed as shown in bottom of Figure 6. The summarization option screen allows a user to specify the summarization compression ratio. Figure 7 shows the summarization result for four URLs with the compression ratio set as 30%.

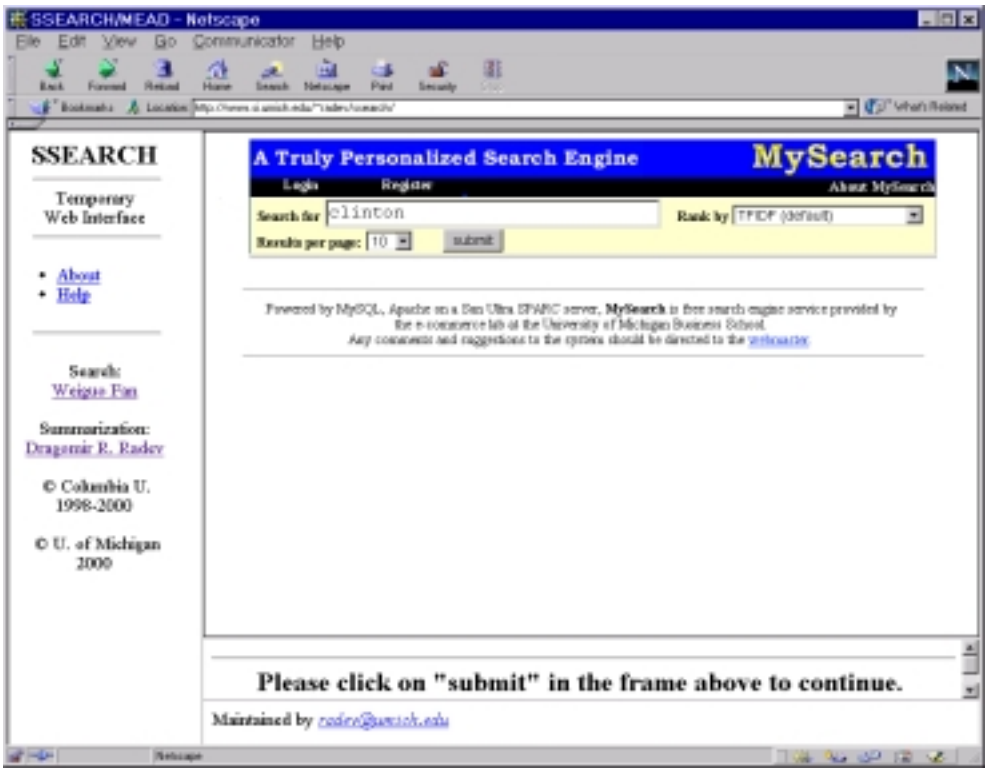

Figure 4: SNS interface (framed)

Figure 5: Search output along with user selection of documents to be summarized

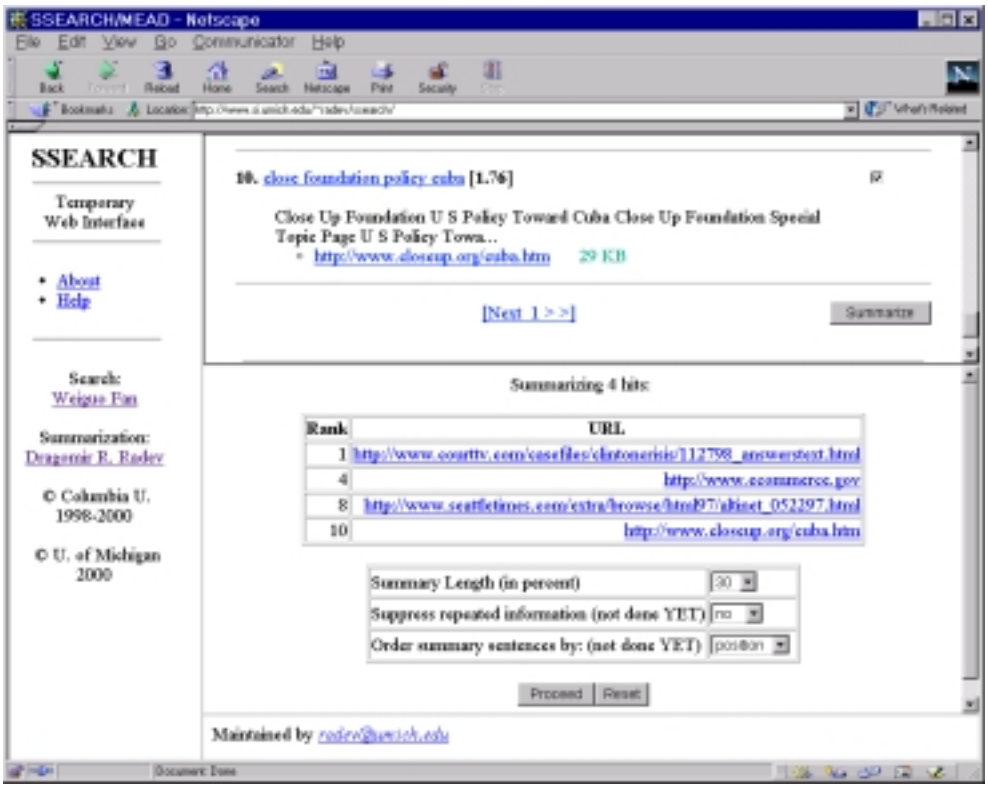

Figure 6: Selected documents for summarization

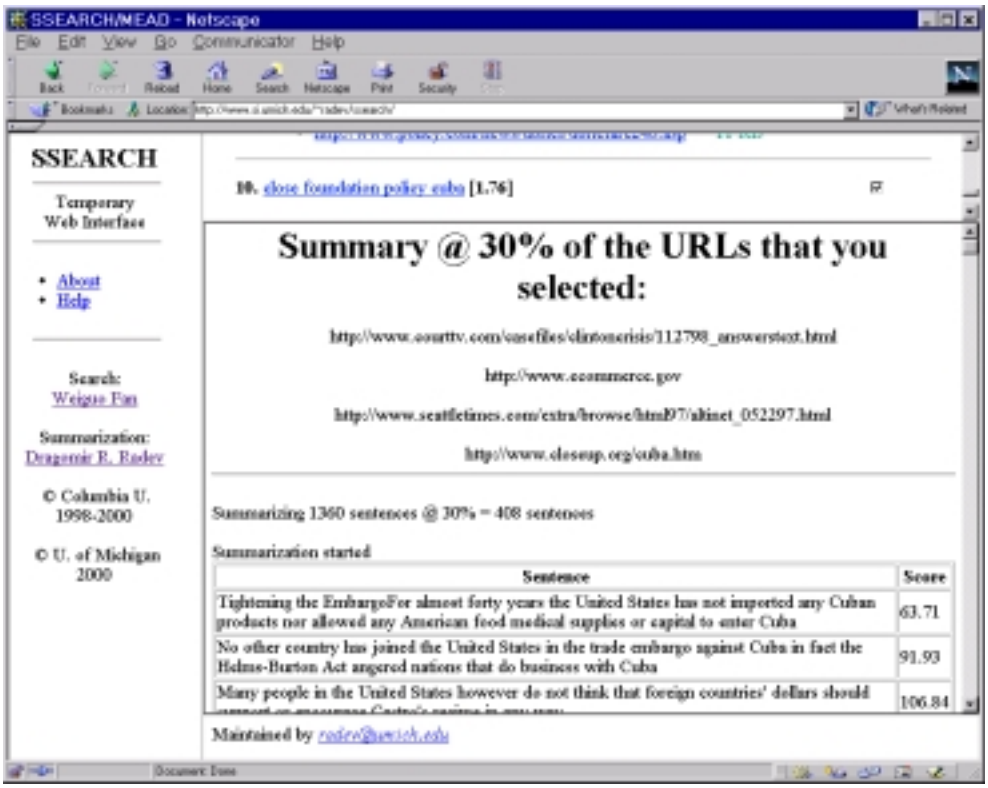

Figure 7: Output of the summarizer

The following information is shown in the summarization result screen in Figure 7:

- The number of sentences in the text of the set of URLs that the user selected
- The number of sentences in the summary

The sentences representing the themes of those selected URLs and their relative scores. The sentences are ordered the same way they appear in the original set of documents.

6 Experimental results

Our system was evaluated using the task-based extrinsic measure as suggested in (Mani et al. 1999). The experiment was set up as follows:

Three sets of documents on different topics were selected prior to the experiment. The topics and their corresponding document information are shown in Table 1.

Topic No.	Topic	No. of Articles	Length
S1	Global E-Commerce Framework	3	200k
S2	Introduction to Data Mining	2	100k
S3	Intelligent Agents and their application in Information retrieval	5	160k

Table 1: Evaluation Topics and their corresponding document set information

Sentence	Score
The idea behind data mining then is the non-trivial process of identifying valid novel potentially useful and ultimately understandable patterns in data 18 2 The term knowledge discovery in databases KDD was formalized in 1989 in reference to the general concept of being broad and 'high level' in the pursuit of seeking knowledge from data	494.92
The term data mining is then this high-level application techniques / tools used to present and analyze data for decision makers	509.11
This term data mining has been used by statisticians data analyst and the MIS management information systems community whereas KDD has been mostly used by artificial intelligence and machine learning researchers	487.92
These are : -the untapped value in large databases consolidation of database records tending towards a single customer view concept of an information or data warehouse from the consolidation of databases dramatic drop in the cost/performance ratio of hardware systems - for data storage and processing	576.60
Intense competition in an increasing saturated marketplace the ability to custom manufacture market and advertise to small market segments and individuals 4 and the market for data mining products is estimated at about 500 million in early 1994 12 Data mining technologies are characterized by intensive computations on large volumes of data	486.92
Data mining versus traditional database queries Traditional database queries contrasts with data mining since these are typified by the simple question such as what were the sales of orange juice in January 1995 for the Boston area	520.53
Data mining on the other hand through the use of specific algorithms or search engines attempts to source out discernable patterns and trends in the data and infers rules from these patterns	500.80

Figure 8: A sample of the summarization result for S2 at 10% compression rate

As Table 1 shows, the articles in topic set S1 are longer than both these in S2 and S3. The articles in S3 are the shortest, with each 32k in average. The number of documents in each topic set is

also different. The variations of document length and different number of documents in each topic set will help test the robustness of our summarization algorithms.

We used SNS to generate both 10% and 20% summaries for each topic. A sample of the 10% summary for topic S2 is shown in Figure 8. Four users were selected for evaluation of these summarization results. Each user was asked to read through the set of full articles for each topic first, followed by its corresponding 10% and 20% summaries. After these 4 users finished each set, they were asked to assign a readability score (1-10) for each summary. The higher the readability score is, the more readable and meaningful for comprehension is the summary. The time of reading both full articles and summaries was tracked and recorded.

	User 1		*User 2*		*User 3*		*User 4*	
Item	Time (Mins)	Readability (1-10)	Time (Mins)	Readability (1-10)	Time (Mins)	Readability (1-10)	Time (Mins)	Readability (1-10)
1: Global E-Commerce ramework (3 articles)	75	N/A	55	N/A	70	N/A	65	N/A
1: 10% Summary	15	9	7	8	10	8	8	7
1: 20% Summary	20	8	12	9	16	7	15	8
2: Introduction to Data ining (2 articles)	55	N/A	42	N/A	49	N/A	46	N/A
2: 10% Summary	10	9	6	8	7	8	6	7
2: 20% Summary	14	8	10	9	12	9	11	8
3: Intelligent Agents nd their application in nformation retrieval (5 rticles)	70	N/A	60	N/A	68	N/A	66	N/A
3: 10% Summary	13	8	7	8	8	7	8	8
3: 20% Summary	20	9	12	9	14	8	15	9

Table 2: Summarization evaluation: detailed results

	10% Summaries	**20% Summaries**
Speedup in reading time by summary over full article	721 / 105 = 6.87	721 / 171= 4.22
Avg. Readability	7.92	8.42

Table 3: Summary of the evaluation results

The detailed evaluation results are shown in Table 2. Table 3 gives the summary of the Table 2. It's shown in Table 2 that these four users have different reading speeds. However, their reading speed is pretty consistent across the 3 topics. The summaries generated by SNS are also very readable. For example, The average readability score (which is obtained by averaging the readability scores assigned by the four users) for 10% and 20% summaries for

topic S1, is 8, 8 respectively. For topic S3, the average readability score for 10% and 20% summaries is 7.75, and 8.75, respectively. Similarly, for S2 the average readability score for 10% and 20% summaries is 8 and 8.5, respectively. The differences in the average readability score also suggest that (a) our summarizer favors longer documents over shorter documents; (b) 20% summaries are generally favorable over 10% summaries. The difference in the readability score between 10% and 20% summaries is bigger in S3 (diff = 1.0) than in S1 (diff = 0). These interesting findings raise interesting questions for future research.

As can be seen from Table 3, the 20% summary achieves better readability score in overall than the 10% summary. The speedup of the 10% summary over full articles is 6.87. That is, with reading material reduced by 900%, the speedup in reading is only 687%. This suggests that there may be a little bit difficulty in reading the 10% summary result. This may be due to the simple sentence boundary detection algorithm we used. The feedback from users in the evaluation seems to confirm the above reason. As more sentences were included in the 20% summaries, the speedup in reading (4.22) almost approached the optimal speedup ratio (5.0)[1].

7 Related Work

Neto et al. (2000) describes a text mining tool that performs document clustering and text summarization. They used the Autoclass algorithm to perform document clustering and used TF-ISF (an adaptation of TF-IDF) to perform sentence ranking and generate the summarization output. Our work is different from theirs in that we perform personalized summarization based on the retrieval result from a generic personalized web-based search engine. A more complicated sentence ranking functions is employed to boost the ranking performance. The compression ratio for the summary is customizable by a user. Both single-document for a single URL and multiple-document summarization for a cluster of URLs are supported in our system.

More related work can be found in Extractor web site http://extractor.iit.nrc.ca/. They use MetaCrawler to perform web-based search and automatically generate summaries for each URLs retrieved. They only support single document summarization in their engine and the compression rate of the summarizer is also non-customizable. We not only support both single and multiple document summarization, but also allow the user to specify the summarization compression ratio as well as to get per-cluster summaries of automatically generated clusters, which, we believe, are more valuable to online users and give them more flexibility and control of the summarization results.

8 Conclusion and Future Work

We described in this paper a prototype system SNS, which integrates natural language processing and information retrieval techniques to perform automatic customized summarization of search engine results. The user interface and detailed design of SNS's components are also discussed. Task-based extrinsic evaluation showed that the system is of reasonably high quality.

The following issues will be addressed in the future.

8.1 Interaction between sentence inclusion in a summary

There are two types of interaction (or reinforcement) between sentences in a summary: negative and positive.

Negative interaction occurs when the inclusion of one sentence in the summary indicates that another sentence should **not** appear in the summary. This is particularly relevant to multi-document summarization as in this case: negative interaction models the non-inclusion of redundant information.

The case of positive interaction involves positive reinforcement between sentences. For example, if a sentence with a referring expression is to be

[1] Since the length of the summary is only 20% of the original documents, the maximum speedup in terms of reading time is 1/0.2=5.

included in a summary, typically the sentence containing the antecedent should also be added.

We will investigate specific setups in which positive and/or negative reinforcement between sentences is practical and useful.

8.2 Personalization

We will investigate additional techniques for producing personalized summaries. Some of the approaches that we are considering are:

- Query words: favoring sentences that include words from the user query in the Web-based scenario
- Personal preferences and interaction history: we would favor sentences that match the user profile (e.g., overlapping with his or her long-term interests and/or recent queries logged by the system).

8.3 Technical limitations

The current version of our system uses a fairly basic sentence delimiting component. We will investigate the user of robust sentence boundary identification modules in the future.
We will also investigate the possibility of some limited-form anaphora resolution component.

8.4 Availability

A demonstration version of SNS is available at the following URL:
http://www.si.umich.edu/~radev/ssearch/

References

Carbonell, J. and Goldstein, J. (1998). *The use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries*. Poster Session, SIGIR'98, Melbourne, Australia.

Censorware (2000). http://www.censorware.org/web_size/.

Extractor (2000). http://extractor.iit.nrc.ca/.

IDS (2000). Internet Domain Survey. http://www.isc.org/ds/.

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

Lawrence, S., and Giles, C. L. (1997). *Searching the World Wide Web*, Science, 280(3), 98-100.

Lawrence, S., and Giles, C. L. (1999). *Accessibility of information on the web*, Nature, 400, 107-109.

Mani, I. and Bloedorn, E. (1999). *Summarizing similarities and differences among related documents*. Information Retrieval 1(1): 35—67.

Mani, I., House, D., Klein, G., Hirschman, L., Obrst, L., Firmin, T., Chrzanowski, M., and Sundheim, B. (1998). *The TIPSTER SUMMAC Text Summarization Evaluation*. The MITRE Corporation Technical Report MTR 98W0000138, McLean, Virginia.

McKeown, K. and D. R. Radev. *Generating Summaries of Multiple News Articles*. Proceedings, ACM Conference on Research and Development in Information Retrieval SIGIR'95 (Seattle, WA, July 1995).

NetSizer (2000). http://www.netsizer.com/.

Neto, J. L., Santos, A. D., Kaestner, C. A. A., and Freitas, A. A. (2000). *Document clustering and text summarization*. In Proceedings, 4th Int. Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000), 41-55. London: The Practical Application Company.

Radev, D. R., Hatzivassiloglou, V., and McKeown, K. *A Description of the CIDR System as Used for TDT-2*. Proceedings, DARPA Broadcast News Workshop, (Herndon, VA, February 1999).

Radev, D. R, Jing, H., and Stys-Budzikowska, M. *Summarization of multiple documents: clustering, sentence extraction, and evaluation*. Proceedings, ANLP-NAACL Workshop on Automatic Summarization, (Seattle, WA, April 2000)

Salton, G. (1989). *Automatic Text Processing*. Addison-Wesley Publishing Co., Reading, MA, 1989.