Trainable, Scalable Summarization Using Robust NLP and Machine Learning*

Chinatsu Aone†, Mary Ellen Okurowski‡, James Gorlinsky†

†SRA International
4300 Fair Lakes Court
Fairfax, VA 22033
{aonec, gorlinsk}@sra.com

‡Department of Defense
9800 Savage Road
Fort Meade, MD 20755-6000
meokuro@afterlife.ncsc.mil

Abstract

We describe a trainable and scalable summarization system which utilizes features derived from information retrieval, information extraction, and NLP techniques and on-line resources. The system combines these features using a trainable feature combiner learned from summary examples through a machine learning algorithm. We demonstrate system scalability by reporting results on the best combination of summarization features for different document sources. We also present preliminary results from a task-based evaluation on summarization output usability.

1 Introduction

Frequency-based (Edmundson, 1969; Kupiec, Pedersen, and Chen, 1995; Brandow, Mitze, and Rau, 1995), knowledge-based (Reimer and Hahn, 1988; McKeown and Radev, 1995), and discourse-based (Johnson et al., 1993; Miike et al., 1994; Jones, 1995) approaches to automated summarization correspond to a continuum of increasing understanding of the text and increasing complexity in text processing. Given the goal of machine-generated summaries, these approaches attempt to answer three central questions:

- How does the system count words to calculate worthiness for summarization?
- How does the system incorporate the knowledge of the domain represented in the text?
- How does the system create a coherent and cohesive summary?

Our work leverages off of research in these three approaches and attempts to remedy some of the difficulties encountered in each by applying a combination of information retrieval, information extraction, and NLP techniques and on-line resources with machine learning to generate summaries. Our DimSum system follows a common paradigm of sentence extraction, but automates acquiring candidate knowledge and learns what knowledge is necessary to summarize.

We present how we automatically acquire candidate features in Section 2. Section 3 describes our training methodology for combining features to generate summaries, and discusses evaluation results of both batch and machine learning methods. Section 4 reports our task-based evaluation.

2 Extracting Features

In this section, we describe how the system counts linguistically-motivated, automatically-derived words and multi-words in calculating worthiness for summarization. We show how the system uses an external corpus to incorporate domain knowledge in contrast to text-only statistics. Finally, we explain how we attempt to increase the cohesiveness of our summaries by using name aliasing, WordNet synonyms, and morphological variants.

2.1 Defining Single and Multi-word Terms

Frequency-based summarization systems typically use a single word string as the unit for counting frequency. Though robust, such a method ignores the semantic content of words and their potential membership in multi-word phrases and may introduce noise in frequency counting by treating the same strings uniformly regardless of context.

Our approach, similar to (Tzoukerman, Klavans, and Jacquemin, 1997), is to apply NLP tools to extract multi-word phrases automatically with high accuracy and use them as the basic unit in the summarization process, including frequency calculation. Our system uses both text statistics (term frequency, or *tf*) and corpus statistics (inverse document frequency, or *idf*) (Salton and McGill, 1983) to derive *signature words* as one of the summarization features. If single words were the sole basis of counting for our summarization application, noise would be

*We would like to thank Jamie Callan for his help with the INQUERY experiments.

introduced both in term frequency and inverse document frequency.

First, we extracted two-word noun collocations by pre-processing about 800 MB of L.A. Times/Washington Post newspaper articles using a POS tagger and deriving two-word noun collocations using mutual information. Secondly, we employed SRA's NameTag^{TM} system to tag the aforementioned corpus with names of people, entities, and places, and derived a baseline database for *tf*idf* calculation. Multi-word names (e.g., "Bill Clinton") are treated as single tokens and disambiguated by semantic types in the database.

2.2 Acquiring Knowledge of the Domain

Knowledge-based summarization approaches often have difficulty acquiring enough domain knowledge to create conceptual representations for a text. We have automated the acquisition of *some* domain knowledge from a large corpus by calculating *idf* values for selecting signature words, deriving collocations statistically, and creating a word association index (Jing and Croft, 1994).

2.3 Recognizing Sources of Discourse Knowledge through Lexical Cohesion

Our approach to acquiring sources of discourse knowledge is much shallower than those of discourse-based approaches. For a target text for summarization, we tried to capture lexical cohesion of signature words through name aliasing with the NameTag tool, synonyms with WordNet, and morphological variants with morphological pre-processing.

3 Combining Features

We experimented with combining summarization features in two stages. In the first batch stage, we experimented to identify what features are most effective for signature words. In the second stage, we took the best combination of features determined by the first stage and used it to define "high scoring signature words." Then, we trained DimSum over high-score signature word feature, along with conventional length and positional information, to determine which training features are most useful in rendering useful summaries. We also experimented with the effect of training and different corpora types.

3.1 Batch Feature Combiner

3.1.1 Method

In DimSum, sentences are selected for a summary based upon a score calculated from the different combinations of signature word features and their expansion with the discourse features of aliases, synonyms, and morphological variants. Every token in a document is assigned a score based on its *tf*idf* value. The token score is used, in turn, to calculate the score of each sentence in the document. The score of a sentence is calculated as the average of the scores of the tokens contained in that sentence. To obtain the best combination of features for sentence extraction, we experimented extensively.

The summarizer allows us to experiment with both how we count and what we count for both inverse document frequency and term frequency values. Because different baseline databases can affect *idf* values, we examined the effect on summarization of multiple baseline databases based upon multiple definitions of the signature words. Similarly, the discourse features, i.e., synonyms, morphological variants, or name aliases, for signature words, can affect *tf* values. Since these discourse features boost the term frequency score within a text when they are treated as variants of signature words, we also examined their impact upon summarization.

After every sentence is assigned a score, the top n highest scoring sentences are chosen as a summary of the content of the document. Currently, the DimSum system chooses the number of sentences equal to a power k (between zero and one) of the total number of sentences. This scheme has an advantage over choosing a given percentage of document size as it yields more information for longer documents while keeping summary size manageable.

3.1.2 Evaluation

Over 135,000 combinations of the above parameters were performed using 70 texts from L.A. Times/Washington Post. We evaluated the summary results against the human-generated extracts for these 70 texts in terms of F-Measures. As the results in Table 1 indicate, name recognition, alias recognition and WordNet (for synonyms) all make positive contributions to the system summary performance.

The most significant result of the batch tests was the dramatic improvement in performance from withholding person names from the feature combination algorithm.The most probable reason for this is that personal names usually have high *idf* values, but they are generally not good indicators of topics of articles. Even when names of people are associated with certain key events, documents are not usually *about* these people. Not only do personal names appear to be very misleading in terms of signature word identification, they also tend to mask synonym group performance. WordNet synonyms appear to be effective only when names are suppressed.

3.2 Trainable Feature Combiner

3.2.1 Method

With our second method, we developed a trainable feature combiner using Bayes' rule. Once we had defined the best feature combination for high scoring *tf*idf* signature words in a sentence in the first round, we tested the inclusion of commonly acknowledged positional and length informa-

Entity	Place	Person	Alias	Syn.	F-M
+	+	-	+	+	41.3
+	+	-	-	+	40.7
+	+	-	+	-	40.4
+	+	-	-	-	39.6
-	-	-	-	+	39.5
-	-	-	-	-	39.0
+	+	+	-	-	37.4
+	+	+	+	+	37.4
+	+	+	+	-	37.2
+	+	+	-	+	36.7

Table 1: Results for Different Feature Combinations

tion. From manually extracted summaries, the system automatically learns to combine the following extracted features for summarization:

- short sentence length (less than 5 words)
- inclusion high-score *tf*idf* signature words in a sentence
- sentence position in a document (1st, 2nd, 3rd or 4th quarter)
- sentence position in a paragraph (initial, medial, final)

Inclusion in the high scoring *tf*idf* signature word set was determined by a variable system parameter (identical to that used in the pre-trainable version of the system). Unlike Kupiec *et al.*'s experiment, we did not use the *cue word* feature. Possible values of the paragraph feature are identical to how Kupiec *et al.* used this feature, but applied to all paragraphs because of the short length of the newspaper articles.

3.2.2 Evaluation

We performed two different rounds of experiments, the first with newspaper sets and the second with a broader set from the TREC-5 collection (Harman and Voorhees, 1996). In both rounds we experimented with

- different feature sets
- different data sources
- the effects of training.

In the first round, we trained our system on 70 texts from the L.A. Times/Washington Post (latwp-dev1) and then tested it against 50 new texts from the L.A. Times/Washington Post (latwp-test1) and 50 texts from the Philadelphia Inquirer (pi-test1). The results are shown in Table 2. In both cases, we found that the effects of training increased system scores by as much as 10% F-Measure or greater. Our results are similar to those of Mitra (Mitra, Singhal, and Buckley, 1997), but our system with the trainable combiner was able to outperform the lead sentence summaries.

Text Set	Training?	F-M	Lead
latwp-dev1	NO	41.3	
latwp-dev1	YES	49.9	48.2
latwp-test1	NO	31.9	
latwp-test1	YES	44.6	42.0
pi-test1	NO	40.5	
pi-test1	YES	49.7	47.7

Table 2: Results on Different Test Sets with or without Training

F-M	Sentence Length	High Score	Document Position	Paragraph Position
24.6	-	-	-	+
24.6	+	-	-	+
39.2	+	-	-	-
39.7	-	-	-	-
39.7	-	+	-	-
39.7	+	+	-	-
39.7	-	+	-	+
39.7	+	+	-	+
43.8	-	-	+	-
45.1	-	-	+	+
45.5	+	-	+	+
45.7	+	-	+	-
46.6	-	+	+	-
46.6	+	+	+	-
48.4	-	+	+	+
49.9	+	+	+	+

Table 3: Effects of Different Training Features

Table 3 summarizes the results of using different training features on the 70 texts from L.A. Times/Washington Post (latwp-dev1). It is evident that positional information is the most valuable, while the sentence length feature introduces the most noise. High scoring signature word sentences contribute, especially in conjunction with the positional information and the paragraph feature. High Score refers to using an *tf*idf* metric with WordNet synonyms and name aliases enabled, person names suppressed, but all other name types active.

The second round of experiments were conducted using 100 training and 100 test texts for each of six sources from the the TREC 5 corpora (i.e., Associated Press, Congressional Records, Federal Registry, Financial Times, Wall Street Journal, and Ziff). Each corpus was trained and tested on a large baseline database created by using multiple text sources. Results on the test sets are shown in Table 4. The discrepancy in results among data sources suggests that summarization may not be equally viable for all data types. This squares with results reported in (Nomoto and Matsumoto, 1997) where learned attributes varied in effectiveness by text type.

Text Set	F-M	Precision	Recall	Short	High Score	Doc. Position	Para. Position
ap-test1	49.7	47.5	52.1	YES	YES	YES	YES
cr-test1	36.1	35.1	37.0	YES	NO	YES	YES
fr-test1	38.4	33.8	44.5	YES	NO	YES	YES
ft-test1	46.5	41.8	52.3	YES	YES	YES	NO
wsj-test1	51.5	48.5	54.8	YES	NO	YES	YES
zf-test1	46.6	45.0	48.3	NO	YES	YES	YES

Table 4: Results of Summaries for Different Corpora

4 Task-based Evaluation

The goal of our task-based evaluation was to determine whether it was possible to retrieve automatically generated summaries with similar precision to that of retrieving the full texts. Underpinning this was the intention to examine whether a generic summary could *substitute* for a full-text document given that a common application for summarization is assumed to be browsing/scanning summarized versions of retrieved documents. The assumption is that summaries help to accelerate the browsing/scanning without information loss.

Miike *et al.* (1994) described preliminary experiments comparing browsing of original full texts with browsing of dynamically generated abstracts and reported that abstract browsing was about 80% of the original browsing function with precision and recall about the same. There is also an assumption that summaries, as encapsulated views of texts, may actually improve retrieval effectiveness. (Brandow, Mitze, and Rau, 1995) reported that using programmatically generated summaries improved precision significantly, but with a dramatic loss in recall.

We identified 30 TREC-5 topics, classified by the easy/hard retrieval schema of (Voorhees and Harman, 1996), five as hard, five as easy, and the remaining twenty were randomly selected. In our evaluation, INQUERY (Allan et al., 1996) retrieved and ranked 50 documents for these 30 TREC-5 topics. Our summary system summarized these 1500 texts at 10% reduction, 20%, 30%, and at what our system considers the BEST reduction. For each level of reduction, a new index database was built for INQUERY, replacing the full texts with summaries.

The 30 queries were run against the new database, retrieving 10,000 documents per query. At this point, some of the summarized versions were dropped as these documents no longer ranked in the 10,000 per topic, as shown in Table 5. For each query, all results except for the documents summarized were thrown away. New rankings were computed with the remaining summarized documents. Precision for the INQUERY baseline (INQ.base) was then compared against each level of the reduction. Table 6 shows that at each level of reduction the overall precision dropped for the summarized versions. With more reduction, the drop was more dramatic. However, the BEST summary version performed better than the percentage methods.

Precision at	INQ.base	INQ.BEST
5 docs	.8000	.8000
10 docs	.8000	.7800
15 docs	.7465	.7200
20 docs	.7600	.7200
30 docs	.7067	.6733

Table 7: Precision for 5 High Recall Queries

We examined in more detail document-level averages for five "easy" topics for which the INQUERY system had retrieved a high number of texts. Table 7 reveals that for topics with a high INQUERY retrieval rate the precision is comparable. We posit that when queries have a high number of relevant documents retrieved, the summary system is more likely to *reduce* information rather than *lose* information. Query topics with a high retrieval rate are likely to have documents on the subject matter and therefore the summary just reduces the information, possibly alleviating the browsing/scanning load.

We are currently examining documents lost in the re-ranking process and are cautious in interpreting results because of the difficulty of closely correlating the term selection and ranking algorithms of automatic IR systems with human performance. Our experimental results do indicate, however, that generic summarization is more useful when there are many documents of interest to the user and the user wants to scan summaries and weed out less relevant document quickly.

5 Summary

Our summarization system leverages off research in information retrieval, information extraction, and NLP. Our experiments indicate that automatic summarization performance can be enhanced by discovering different combinations of features through a machine learning technique, and that it can exceed lead summary performance and is affected by data source type. Our task-based evaluation reveals that generic summaries may be more effectively applied to high-recall document retrievals.

Run	INQ.base	INQ.10%	INQ.20%	INQ.30%	INQ.BEST
Retrieved	1500	1500	1500	1500	1500
Relevant	4551	4551	4551	4551	4551
Rel-ret	415	294 (-29.2%)	332 (-20.0%)	335 (-19.3%)	345 (-16.9%)

Table 5: INQUERY Baseline Recall vs. Summarized Versions

Precision at	INQ.base	INQ.10%	INQ.20%	INQ.30%	INQ.BEST
5 docs	0.4133	0.3267 (-21.0)	0.3800 (- 8.1)	0.3067 (-25.8)	0.3333 (-19.4)
10 docs	0.3700	0.2600 (-29.7)	0.2800 (-24.3)	0.2933 (-20.7)	0.3100 (-16.2)
15 docs	0.3511	0.2400 (-31.6)	0.2800 (-20.3)	0.2867 (-18.3)	0.2867 (-18.3)
20 docs	0.3383	0.2217 (-34.5)	0.2600 (-23.1)	0.2733 (-19.2)	0.2717 (-19.7)
30 docs	0.3067	0.2056 (-33.0)	0.2400 (-21.7)	0.2522 (-17.8)	0.2556 (-16.7)

Table 6: INQUERY Baseline Precision vs. Summarized Versions

References

Allan, J., J. Callan, B. Croft, L. Ballesteros, J. Broglio, J. Xu, and H. Shu Ellen. 1996. Inquery at trec-5. In *Proceedings of The Fifth Text REtrieval Conference (TREC-5).*

Brandow, Ron, Karl Mitze, and Lisa Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31:675–685.

Edmundson, H. P. 1969. New methods in automatic abstracting. *Journal of the Association for Computing Machinery*, 16(2):264–228.

Harman, Donna and Ellen M. Voorhees, editors. 1996. *Proceedings of The Fifth Text REtrieval Conference (TREC-5).* National Institute of Standards and Technology, Department of Commerce.

Jing, Y. and B. Croft. 1994. *An Association Thesaurus for Information Retrieval.* Technical Report 94-17. Center for Intelligent Information Retrieval, University of Massachusetts.

Johnson, F. C., C. D. Paice, W. J. Black, and A. P. Neal. 1993. The application of linguistic processing to automatic abstract generation. *Journal of Documentation and Text Management*, 1(3):215–241.

Jones, Karen Sparck. 1995. Discourse modeling for automatic summaries. In E. Hajicova, M. Cervenka, O. Leska, and P. Sgall, editors, *Prague Linguistic Circle Papers*, volume 1, pages 201–227.

Kupiec, Julian, Jan Pedersen, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th Annual International SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73.

McKeown, Kathleen and Dragomir Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International SIGIR Conference on Research and Development in Information*, pages 74–78.

Miike, Seiji, Etsuo Itho, Kenji Ono, and Kazuo Sumita. 1994. A full text retrieval system with a dynamic abstract generation function. In *Proceedings of 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 152–161.

Mitra, Mandar, Amit Singhal, and Chris Buckley. 1997. An Automatic Text Summarization and Text Extraction. In *Proceedings of Intelligent Scalable Text Summarization Workshop, Association for Computational Linguistics (ACL)*, pages 39–46.

Nomoto, T. and Y. Matsumoto. 1997. Data reliability and its effects on automatic abstraction. In *Proceedings of the Fifth Workshop on Very Large Corpora.*

Reimer, Ulrich and Udo Hahn. 1988. Text condensation as knowledge base abstraction. In *Proceedings of the 4th Conference on Artificial Intelligence Applications (CAIA)*, pages 338–344.

Salton, G. and M. McGill, editors. 1983. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Co., New York, New York.

Tzoukerman, E., J. Klavans, and C. Jacquemin. 1997. Effective use of naural language processing techniques for automatic conflation of multi-word terms: the role of derivational morphology, part of speech tagging and shallow parsing. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development of Information Retrieval*, pages 148–155.

Voorhees, Ellen M. and Donna Harman. 1996. Overview of the fifth text retrieval conference (trec-5). In *Proceedings of The Fifth Text REtrieval Conference (TREC-5).*