Summarization Techniques at DUC 2004

Yllias Chali   Maheedhar Kolla
Department of Mathematics and Computer Science
University of Lethbridge
4401 University Drive
Lethbridge, Alberta, Canada, T1K 3M4
E-mail: {chali, kolla}@cs.uleth.ca

Abstract

This paper presents the summarization techniques implemented by the University of Lethbridge summarizer in order to generate very short summary (<= 75 bytes) and short summary (<= 665 bytes) from single and multiple documents. We present these techniques in the context of DUC 2004.

1 Introduction

Document Understanding Conference (DUC) is an evaluation series, organized by National Institute for Standards and Technology (NIST) and supported by Defense Advanced Research Projects Agency (DARPA), in the area of automatic text summarization. NIST selected 50 English document clusters from the TDT collection, 25 Arabic document clusters from the TDT collection and 50 English document clusters from the TREC collection. Each TDT or TREC cluster selected contains on an average 10 documents. Also, NIST provided questions for the 50 TREC clusters. Five tasks were defined in DUC 2004:

Task 1 – Using the 50 TDT English clusters, create a very short summary (<= 75 bytes) for each document in the given cluster.

Task 2 – Using the 50 TDT English clusters, create a short summary (<= 665 bytes) for each cluster.

Task 3 – Two required runs and one optional one per group:

- Required: Using the automatic English translations of each document in the 25 TDT clusters, create a very short summary (<= 75 bytes) of the document.
- Required: Using the manual English translation of each document in the 25 TDT clusters, create a very short summary (<= 75 bytes) of the document.
- Optional: Using the MT output and any other documents from English sources, e.g., relevant documents for the 25 TDT clusters provided by NIST.

Task 4 – Two required runs and one optional one per group:

- Required: Using the given one or more automatic English translations for each document in the 25 TDT document clusters, create a short summary (<= 665 bytes) for the cluster in English.

- Required: Using the given manual English translation for each document in the 25 TDT clusters, create a short summary (<= 665 bytes) for the cluster in English.
- Optional: Using the MT output and any other documents from English sources, e.g., relevant documents for the 25 TDT clusters provided by NIST.

Task 5 – Using the 50 TREC clusters with each document cluster having a corresponding question of the form “who is X?”, where X is the name of a person or group of people, create a short summary (<= 665 bytes) of the cluster as a response to the question.

Our system took part in three tasks (task 1, 2 and 5). In section 2 we briefly discuss about the various stages that are common to both single document summarization and multi-document summarization. In section 3 we describe about the extraction stage with regard to the three tasks. We then evaluate the performance results of our system.

2 System Overview

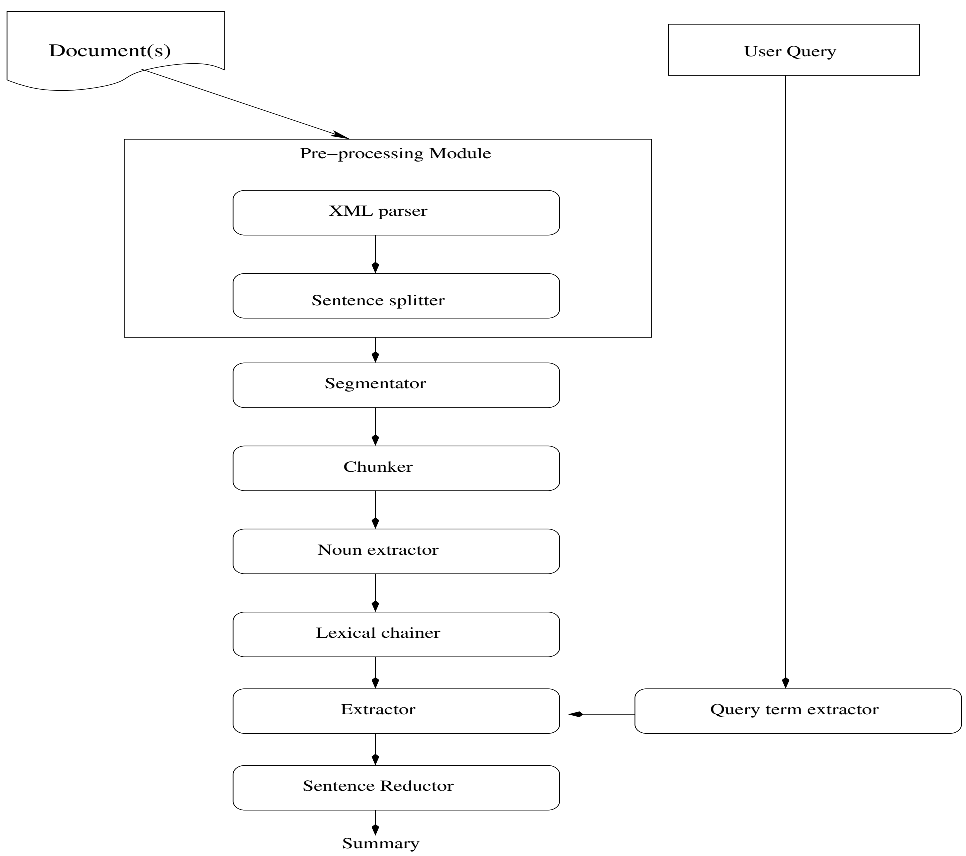

Figure 1: Overview of UofL summarizer

Pre-processing

Pre-processing involves parsing the document(s) using XML parser to extract the textual information and using sentence splitter (Sekine, 2002) to separate the sentences.

Segmentation

In order to identify the topic boundaries in the given input text, we perform linear text segmentation (Choi, 2000). This process of segmentation would be helpful for better analysis of the given text and to generate efficient summary.

Text chunking

Our system utilizes the text chunker (Sekine, 2002) to generate a syntactically correlated representation of each text segment. This representation would be useful to extract the nouns in the following stage.

Noun extraction

Based on the principle that nouns characterize the topic of a particular text segment, we perform noun extraction from the chunker output.

Lexical chains

Our system computes the lexical chains (Morris and Hirst, 1991) as an intermediate representation for each segment. This intermediate representation generated will be useful in order to rank the segments and sentences during the extraction stage. Lexical chains are a sequence of semantically related words spanning a topical unit of the text.

$$LC = \{chainMember_1, chainMember_2, \ldots, chainMember_n\}$$

where $chainMember_i$s are word senses and there exists a semantic relation between any pair of word senses in a given chain.

3 Extraction

This stage differs for the task involving single document summarization (task 1) and for tasks involving multi-document summarization (task 2 and task 5) in the context of DUC 2004.

Segment ranking

Our system ranks the segments, with respect to the lexical chains created, based on the following equation :

$$score(segment_i) = \sum_{j=1}^{m} \frac{score(chainMember_j, segment_i)}{segments_j} \tag{1}$$

where $score(segment_i)$ is the score of $segment_i$, $score(chainMember_j, segment_i)$ is the number of occurrence of the $chainMember_j$ in $segment_i$, and $segments_j$ is the number of segments having the $chainMember_j$.

We then select the top ranked segments for sentence extraction.

Sentence ranking

Our system performs sentence ranking based on the following equation:

$$score(sentence_i) = \sum_{j=1}^{m} \frac{score(chainMember_j, sentence_i)}{sentences_j} \tag{2}$$

where $score(sentence_i)$ is the score of $sentence_i$, $score(chainMember_j, sentence_i)$ is the number of occurrences of the $chainMember_j$ in $sentence_i$, and $sentences_j$ is the number of sentences in that segment in which the $chainMember_j$ occurs.

Text clustering

Text clustering is the method of dividing the collection of documents into a group of clusters based on similarity measure. Our system computes the similarity measure between two segments based on the number of lexical chain members shared by them.

Cluster ranking

Our system ranks the clusters, with regard to the lexical chains, based on the following equation:

$$score(cluster_i) = \sum_{j=1}^{m} \frac{score(chainMember_j, cluster_i)}{clusters_j} \tag{3}$$

where $score(cluster_i)$ is the score of $cluster_i$, $score(chainMember_j, cluster_i)$ is the number of occurrences of $chainMember_j$ in $cluster_i$, and $clusters_j$ is the number of clusters having the $chainMember_j$.

3.1 Single document summarization

Task 1

We perform segment-ranking and sentence ranking in order to identify and extract the top ranked sentences from the whole document. We then generate headline ($<=$ 75 bytes) by applying sentence reduction techniques to the sentences extracted from the sentence-ranking phase.

3.2 Multi-document summarization

Task 2

In this task we perform text clustering. We then extract the sentences by selecting the top ranked sentences, from the top ranked segments from the top ranked clusters. These sentences are then arranged in a chronological order, by sorting them with respect to the time stamps of the documents they are extracted from, generating a summary ($<=$ 665 bytes).

Task 5

In order to generate a multi-document summary with respect to a user's query, the segments are first clustered and the top most ranked cluster is selected. We then extract the sentences, using pattern matching techniques, from the top ranked segments of that cluster.Finally a summary is generated ($<=$ 665 bytes) by arranging the sentences in a chronological order considering the time stamps of the source documents they are extracted from.

ROUGE-N	Mean	95% CI Lower	95% CI Upper
ROUGE-1	0.12062	0.11107	0.13017
ROUGE-2	0.02517	0.02089	0.02945
ROUGE-3	0.00631	0.00454	0.00808
ROUGE-4	0.00119	0.00057	0.00181
ROUGE-L	0.10772	0.09917	0.11627
ROUGE-W-1.2	0.06542	0.06060	0.07024

Table 1: Evaluation results using ROUGE automatic n-gram matching (task 1)

ROUGE-N	Mean	95% CI Lower	95% CI Upper
ROUGE-1	0.30355	0.28573	0.32137
ROUGE-2	0.04747	0.03900	0.05594
ROUGE-3	0.01179	0.00793	0.01565
ROUGE-4	0.00427	0.00195	0.00659
ROUGE-L	0.31606	0.29946	0.33266
ROUGE-W-1.2	0.10782	0.10239	0.11325

Table 2: Evaluation results using ROUGE automatic n-gram matching (task 2)

ROUGE-N	Mean	95% CI Lower	95% CI Upper
ROUGE-1	0.30952	0.29282	0.32622
ROUGE-2	0.06959	0.06054	0.07864
ROUGE-3	0.02611	0.02038	0.03184
ROUGE-4	0.01291	0.00897	0.01685
ROUGE-L	0.32527	0.30900	0.34154
ROUGE-W-1.2	0.11103	0.10569	0.11637

Table 3: Evaluation results using ROUGE automatic n-gram matching (task 5)

Measure	Value
Mean coverage	0.16566
Median coverage	0.052
Sample std of coverage	0.26016
Mean of quality questions	1.90

Table 4: Evaluation results using SEE manual evaluation (task 2)

4 Evaluation

In this section, we show the performance results of our system for the three tasks participated in DUC 2004. Tables (1, 2, 3) show the results of our system performance evaluated by ROUGE automatic n-gram matching (Lin and Hovy, 2003). Tables (4, 5) show the results of our system performance using SEE manual evaluation protocol (Lin, 2001). Our system did better in quality questions for task 2 and showed good results in quality questions, mean coverage, and responsiveness for task 5. Table 6 shows the average category of the summary's quality with respect to each quality question.

Measure	Value
Mean coverage	0.19868
Median coverage	0.048
Sample std of coverage	0.30492
Mean of quality questions	1.82
Responsiveness	1.42

Table 5: Evaluation results using SEE manual evaluation (task 5)

	Task 2	Task 5
Q.No	Mean category	Mean category
Q1	3.28	2.9
Q2	2.7	2.42
Q3	1.36	1.38
Q4	2.34	2
Q5	1.08	1.46
Q6	1.22	1.3
Q7	1.36	1.3

Table 6: Mean of performance with respect to quality questions

5 Conclusion

In this paper, we presented briefly the summarization techniques to generate very short and short summaries of single and multiple documents in context of DUC 2004. Our system performed better in quality question for task 2 and task 5, and responsiveness for task 5 .

Acknowledgments

This work was supported by a research grant from the Natural Sciences and Engineering Research Council of Canada and Research Excellence Envelope funding from the Alberta Heritage Foundation for Science and Engineering Research.

References

Barzilay, R. and M. Elhadad. (1997). Using lexical chains for text summarization. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th European Chapter Meeting of the Association for Computational Linguistics, Workshop on Intelligent Scalable Text Summarization*, pages 10-17, Madrid.

Choi, F. Y. Y. (2000). Advances in domain independent linear text segmentation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics*, pages 26 - 33, Seattle, Washington.

Lin, C. Y., (2001). *Summary Evaluation Environment.* http://www.isi.edu/∼cyl/SEE.

Lin, C. Y. and E. Hovy. (2003). Automatic evaluation os summaries using n-gram co-occurrence statistics. In *Proceedings of Human Technolgy Conference 2003 (HLT-NAACL2003)*, Edmonton, Canada.

Mani, I. (2001). *Automatic Summarization*. John Benjamins.

Mani, I. and M. Maybury. (1999). *Advances in Automatic Text Summarization*. MIT Press.

Morris, J. and G. Hirst. (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21-48.

Sekine, Satoshi, (2002). *Proteus Project - OAK System (English Sentence Analyzer)*. http://nlp.cs.nyu.edu/oak.

Silber, H. G. and K. F. McCoy. (2002). Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496.