KU Text Summarization System for DUC 2003

Kyoung-Soo Han **Young-In Song** **Hae-Chang Rim**
Dept. of Computer Science and Engineering
Korea University
1, 5-ga, Anam-dong, Seoul 136-701, Korea
{kshan, sprabbit, rim}@nlp.korea.ac.kr

Abstract

We present the KU multi-document summarization system which is designed for performing task 1, task 2, and task 3 of DUC 2003. The summarizer is composed of information extraction, information normalization, information fusion, and summary generation modules. At present, the performance of the system is not good because of the incomplete merging and generation modules.

1 Introduction

This paper describes our multi-document summarization system for DUC 2003. As a first participation of DUC, our goal of this year is to design a prototype of a multi-document summarization system which can be expanded and improved in the future.

Given a single document or multi documents to be summarized, optionally with a topic or a viewpoint, our summarizer generates a summary(*abstract*) for the document(s) beyond *extract*.

We consider the summarization task as extracting, condensing, and re-expressing the information in the documents. From this point of view, a summarizer (1)extracts information from the input document(s), and represents it with a knowledge representation technique, (2)filters out the redundant information and merge the information, and (3)generates a summary from the arranged information. Each stage can be related to interpretation, transformation, or generation process stated in (Jones, 1999).

From the next section, we describe the proposed system, experimentation, and the future works in detail.

2 Proposed Summarizer

As represented in figure 1, our system is composed of following four modules: important information extraction module, information normalization module, information fusion module, and summary generation module. The first two modules correspond to the surface representation structuring stage, and the third and fourth modules correspond to merging stage and generation stage respectively.

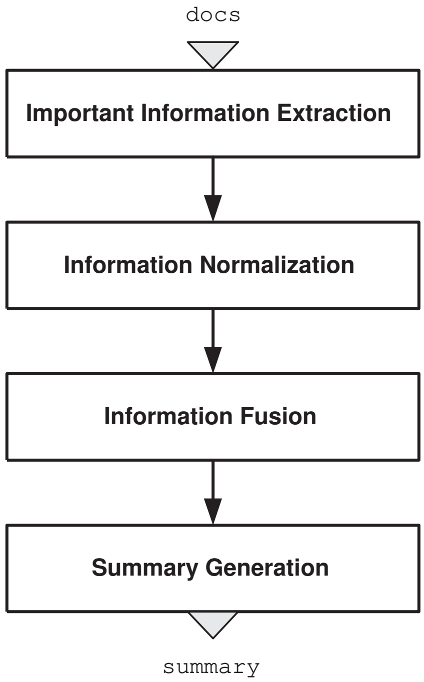

Figure 1: System Architecture

2.1 Important Information Extraction

A document contains a lot of information, all of which are not closely related to the document topic. In order to filter out off-topic information and reduce the amount of information processed at the subsequent steps, we need to extract on-topic information from a document. The

(a) He died at his home in Santa Clara, Calif.

(b) (die, he:**subj**, at:**loc**, in:**loc**)
    (at, home:**pcomp**)
        (home, he:**attr**)
    (in, clara:**pcomp**)
        (clara, santa:**attr**, calif:**mod**)

Figure 2: Information Representation Using Dependency Relations

extracted information is used for building a base of interpretation.

In our system, we consider a sentence as a information unit, and extract important sentences from the input based on lexical chain (Song, 2002). The sentence extraction module is modified to perform the topic focused summarization. For constructing lexical chains, all words are not used, but only on-topic words having some relations in WordNet (Fellbaum, 1998) with the topic words are used. As a consequence, on-topic lexical chains are constructed, with which all sentences in a document or a document cluster are weighted and ranked. 30% of highly ranked sentences are extracted. For the details of sentence weighting, refer to (Song, 2002).

2.2 Information Normalization

2.2.1 Information representation

Since the syntactic information is useful for merging the related information between sentences and generating a summary, the syntactic information produced by a syntactic parser is used for a base of internal representation.

Connexor FDG Parser (Tapanainen and Jarvinen, 1997) is used for syntactic parsing. Its output represents dependency functions, morphological tags, functional tags, surface syntactic tags of each word in a sentence. Extracted sentences are parsed and transformed into a set of dependency relations with syntactic information. Figure 2 shows an example of the information representation using dependency relations. Each row in Figure 2 (b) represents a dependency relation between a head word and the modifying word. The words shown in bold type represent the dependency relations, and the attached syntactic information is omitted for convenience.

At this moment, we do not implement any semantic processes, and the syntactic structure including functional relations is used for representing extracted information.

2.2.2 Information standardization

A *complex sentence*, composed of several *simple sentences*, seems too coarse to be a unit of a information representation. Therefore, a complex sentence is segmented into simple sentences, each of which is used as a unit of representation. We will call this simple sentence by *unit sentence* for convenience, and simply refer to unit sentence as a sentence unless its meaning is confused.

Sentences conveying the same information frequently have somewhat different structures. In order to normalize the structures, some standard forms for the various representations are required. We transfer sentences into the standard forms as follows.

- **Participial clause** The participial clause modifying the main verb of a sentence is separately split into a new sentence, and its subject and tense information are reconstructed from the main clause.

- **Relative pronoun clause** The subordinate clause that has a relative pronoun and modifies the main verb of a sentence is separately split into a new sentence, and the relative pronoun is replaced with its antecedent.

- **Reporting verb** The final clause of the reporting verb, such as `say` or `report`, functioning as the main verb of a sentence is separately split into a new sentence, and the reporting verb clause itself is deleted.

- **Passive voice** A passive sentence is converted into an active one.

- **Conjunction** A sentence concatenated with coordinate conjunctions is separately split into several distinct sentences.

2.3 Information Fusion

2.3.1 Unit sentence merging

In this step, sentences representing the same information are identified and merged. Because the input documents are topically related ones, the decision whether two sentences convey the same information or not is simply made based on the predicate of the sentence and its arguments. When sentences, consisting of dependency relations between words, are merged, merging is carried out individually according to the dependency relations.

For example, figure 3 is identified as the sentence conveying the same information as figure 2, because the main verb of each sentence is same. The scattered information is united by this merging step.

(a) He died of colon cancer shortly after retirement from nearly 50 years of drawing every "Peanuts" strip.

```
(b) (die, he:subj, of:cnt, after:tmp)
        (of, cancer:pcomp)
            (cancer, colon:attr)
        (after, shortly:meta, retirement:pcomp)
            (retirement, from: mod)
                (from, year:pcomp)
                    (year, 50:qn, of:mod)
                        (50, nearly:ad)
                        (of, draw:pcomp)
                            (draw, strip:obj)
                                (strip, every:det, peanut:attr)
```

Figure 3: Related information to the figure 2

(a) He died of colon cancer at his home in Santa Clara, Calif. shortly after retirement from nearly 50 years of drawing every "Peanuts" strip.

```
(b) (die, he:subj, of:cnt, at:loc, in:loc, after:tmp)
        (of, cancer:pcomp)
            (cancer, colon:attr)
        (at, home:pcomp)
            (home, he:attr)
        (in, clara:pcomp)
            (clara, santa:attr, calif:mod)
        (after, shortly:meta, retirement:pcomp)
            (retirement, from: mod)
                (from, year:pcomp)
                    (year, 50:qn, of:mod)
                        (50, nearly:ad)
                        (of, draw:pcomp)
                            (draw, strip:obj)
                                (strip, every:det, peanut:attr)
```

Figure 4: Result of merging figure 2 and figure 3

2.3.2 Unit sentence scoring

We need a method of scoring unit sentences to determine which information is useful for generating a summary.

$$Score(s_i) = Score(dep_{top})$$
$$Score(dep_j) = Score(head_j) + \alpha \sum_k Score(mod_{jk})$$
$$Score(w_{mn}) = \frac{tf_{mn}}{max_l tf_{ml}}$$

The score of the sentence s_i is calculated by the score of its top-level dependency which consists of the predicate of the sentence and its arguments. The score of a dependency dep_j is computed as the sum of the score of the head word $head_j$ and that of all modifying words mod_{jk}. The score of modifying words is dampened by α, which is determined empirically. We used 0.5 in our experiments.

If a word w_{mn} in a document d_n has some subordinate dependencies, its score is calculated by recursively computing the dependency score. However, if it has no subordinate dependencies, its score is calculated just based on itself, which is the normalized term frequency.

2.3.3 Dependency selection

Because of the unit sentence merging, it is possible for the main verb of a sentence to have more than one modifying word, *dependent*, with the same dependency relation. Only one of these possible dependents is selected and used for the summary generation. Various methods are possible for this dependency selection. In this experiment, the phrase with the most frequent head word is selected.

2.4 Summary Generation

We generate sentences using the syntactic information attached to the words in our simplified representation to produce an abstract.

A sentence type is determined by the essential constituents in the dependents for the main verb, and the order of the dependents is settled according to the sentence type. The essential constituents we used are subject, subjective complement, object, objective complement, and indirect object. Additionally adverbial phrases of location and temporal are concatenated in order when the target length is allowed.

If the initial syntactic information in the unit sentence is not modified during information normalization and fusion step, a word is realized in the same form as the input word. If there is any modification, the word surface form is generated according to the following additional rules.

- **Verb conjugation** All verbs with modified syntactic information is assumed to be a regular verb.
- **Pronoun declension** The declension of personal pronoun is perfectly covered by the rule.
- **Contraction restoration** The contraction of a negative `not` is restored.

Summary is generated by realizing highly scored unit sentences in the ranked order until the target length is used up.

3 Evaluation

There are 4 tasks in DUC 2003. We take part in task 1, task 2, and task 3 without developing any task specific techniques.

- **Task 1** This task is generic single document summarization. The target length is 10 words. The special techniques are required for generating the very short summary.
- **Task 2** This task is topic-focused multi-document summarization. The target length is 100 words. Given each TDT document cluster and the associated TDT topic, systems have to create a short summary of the cluster.
- **Task 3** This task is user-focused multi-document summarization. The target length is 100 words.

> Given each TREC document cluster and a viewpoint description, systems have to create a short summary of the cluster from the point of view specified. The viewpoint description is a natural language string no larger than a sentence. It describes the important facet(s) of the cluster the assessor has decided to include in the short summary.

We do not distinguish the TDT topic and the TREC viewpoint description, applying the same process to task 2 and task 3.

NIST evaluated summaries for *coverage* and *usefulness* in task 1, and for *quality* and *length-adjusted coverage* in task 2 and task 3. These results of our system are not good. The most plausible reason of the bad performance seems to be the incomplete generation module and the merging module. They are implemented only based on syntactic information. We realize that the syntactic information is not enough to identify the redundant information and to generate a good summary. Furthermore, some sentences are too long to have robust parsing results for the parser we employed.

4 Conclusion

In this paper, we have proposed the KU multi-document summarization system which is designed for performing task 1, task 2, and task 3 of DUC 2003. At present, the system is implemented only based on lexical and syntactic information, and we realize that it is not enough to generate a good summary.

For the future works, we will study on the representation and merge of the semantic information, the anaphora resolution, and the application of the temporal information. The same semantic content can be expressed by the different surface forms or structures. To identify the redundant information in the information fusion stage, we have to somehow use the paraphrasing technique, and employ the conceptual hierarchy such as WordNet.

Currently, our system do not process the anaphora resolution. However, the anaphora resolution process must be included in our system to widely cover the merging candidate and to generate the more readable abstracts. Furthermore, we will have to develop a method of employing temporal information in identifying the same information, and deciding the order of generating abstracts.

References

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. The MIT Press.

Karen Sparck Jones. 1999. Automatic summarizing: factors and directions. In Inderjeet Mani and Mark Maybury, editors, *Advances in Automatic Text Summarization*, pages 1–12. The MIT Press.

Young-In Song. 2002. Automatic summarization based on lexical chains considering word association. Master's thesis, Department of Computer Science and Engineering, Korea University.

Pasi Tapanainen and Timo Jarvinen. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71.