CATS a topic-oriented multi-document summarization system at DUC 2005

Atefeh Farzindar, Frédérik Rozon and Guy Lapalme
RALI-DIRO Université de Montréal
CP 6128. Succ. Centre-ville
Montréal, Québec, Canada, H3C 3J7
{farzinda, rozonfre, lapalme}@iro.umontreal.ca

Abstract

CATS is a multidocument summarizing system developed at the Université de Montréal for DUC2005. From a set of topic related documents, it produces an integrated summary answering the need for information at a given level of granularity. It starts from a thematic analysis of the documents to identify a list of text segments containing interesting aspects related to the subject. It then matches these themes with the ones detected in the question. The very good results obtained at the DUC competition are described and discussed.

1 Introduction

An *Information Synthesis Task* is a type of topic-oriented and informative multi-document summarization to produce a comprehensive and non-redundant report that satisfies a given information need (Amigo et al., 2004).

In this paper, we present CATS, a system for summarizing multiple documents concerning a given topic at a level of granularity specified in a user profile. The system first performs a thematic analysis of the documents and then matches these themes with the ones identified in the question. Once CATS has identified a list of thematic segments containing interesting aspects related to the subject, they are sorted to find the most promising ones. Segments which are too similar to others are removed; they probably came from distinct documents about the same events. In order to improve coherence of the final result, we identified temporal expressions and replaced relative temporal references (e.g. `yesterday`, `next Monday`) by absolute ones such as the date of the event.

This paper is organized as follows: Section 2 describes the sentence extraction method based on the analysis of the cluster of documents and the analysis of relevant topics of the given question. In Section 3, we analyze our results at DUC 2005 and Section 4 concludes this paper.

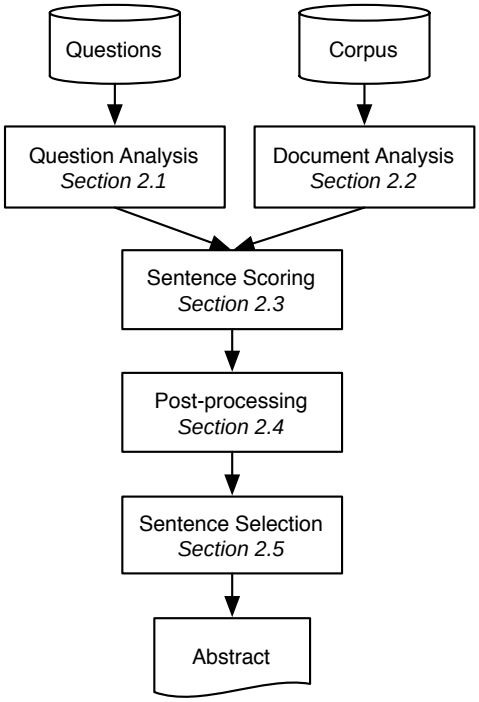

Figure 1: General structure of our system

2 Description of CATS

CATS (*Cats is an Answering Text Summarizer*) uses the extraction of sentences to create a 250-word summary of the cluster. Figure 1 shows the general structure of our system. In the following sections, we detail the 5 steps of CATS processing:

Category	Words in the text
Person	`person, individual, people...`
Organization	`company, organization, corporation...`
Location	`location, position, where, province, country...`
Time	`date, week, day, month, year, time...`

Table 1: Examples of the key words required in the question to identify the 4 categories of named entities considered by our system

2.1 Question analysis

The questions given by the DUC2005 organizers also contain a subject and the expected granularity of the summary (specific or general). The analysis of the questions is done in two steps: the identification of the type of named entities and the splitting of the sentences in basic elements.

2.1.1 Named Entities

Named entities are the most important elements for the generation of a specific summary, which must contain more of this type of information. To increase the precision of their identification in the source texts, we considered 4 categories of named entities for words in the question: *person, organization, location* and *time*. Table 1 gives some examples for each category.

The number of times a word of a category appears in the question increases the probability that this type of named entities must appear in the summary. For example, given the following question, in which we have added manually the identified categories:

`Identify and describe types of organized crime that crosses borders or involves more than one` $\underbrace{\texttt{country}}_{location}$`. Name the` $\underbrace{\texttt{countries}}_{location}$ `involved. Also identify the perpetrators involved with each type of crime, including both` $\underbrace{\texttt{individuals}}_{person}$ `and` $\underbrace{\texttt{organizations}}_{organization}$ `if possible.`

we assign a better score to sentences containing named entities of type *location* and then to sentences containing named entities of type *person* and *organization*.

Head	Mod.	Rel.
`libyans`	`two`	nn
`indicted`	`libyans`	obj
`bombing`	`lockerbie`	nn
`indicted`	`bombing`	for
`bombing`	`1991`	in

Figure 2: Example of decomposition into basic elements of the sentence `two Libyans were indicted for the Lockerbie bombing in 1991`

2.1.2 Basic Elements

A *Basic Element* (Hovy et al., 2005) is a triple (a head, a relationship and a modifier) describing the grammatical relationship between two words in a sentence. Since a basic element does not vary in size, it can be easily compared with another basic element. We use this property to compare the basic elements of the question and the ones of the sentences of the document. Basic elements are also used in one of the many scores of ROUGE.

The module uses the Minipar (Lin, 1998) parser to create the syntactic tree which is then pruned. Once relations between its nodes are resolved, it can result in a list illustrated in figure 2. For CATS, we delete elements that do not seem useful such as determinants (*det*).

This decomposition facilitates sentence comparison because it reduces variation. The result of this step is used as one component of the sentence score.

2.2 Document analysis

Document analysis determines which information is important to include in the summary. The documents are articles from the newspapers Financial Times and Los Angeles Times. Document analysis is done in the following stages after a preprocessing step that splits the text into paragraphs and identifies its publishing date.

2.2.1 Temporal expressions

Newspaper articles often contain temporal expressions that are relative to the publication date; for example, `yesterday, 2 days ago, last month,` etc. Once the sentences are included in our summary, they lose their temporal reference and are no longer meaningful. We solve this problem by calculating, using the TempEx module, the absolute values of these expressions so that they can be understood in any context. The module uses temporal information in the original text, such as the publication date, to resolve the relative temporal expressions. TempEx uses a series of regular expressions to identify these expressions and then it adds TIMEX2 (l. Gerber et al., 2002) markers around them indicating their *absolute* value. An example of a marked-up text is presented in figure 3.

```
At <TIMEX2 TYPE="DATE" VAL="1990" MOD="END">the end of last year</TIMEX2>,
parliament agreed a law banning all publicity for tobacco from <TIMEX2
TYPE="DATE" VAL="199301">January 1993</TIMEX2>, as well as restricting alcohol
advertising to adult newspapers and magazines and a few radio stations.
The government's temptation to curb Seita's freedoms climaxed <TIMEX2
TYPE="DATE" VAL="199104">last month</TIMEX2> when the group was forced to
withdraw its latest brand, Chevignon, after a bitter political row.
```

Figure 3: The result of mark-up performed by TempEx. The markers `TIMEX2`, which are inserted around the temporal expressions, contain the temporal expression type (`TYPE`), its absolute value (`VAL`) and in some cases, a modifier (`MOD`).

Since TempEx needs the original text to resolve the temporal expressions, we execute TempEx as a preprocessing step and store the results for later use.

2.2.2 Thematic segmentation

Newspaper articles often talk about multiple issues. Thematic segmentation determines what sentences pertain to each each topic so that we can focus on those corresponding to the topic of interest. For thematic segmentation, we performed some experiments using two segmenters TextTiling (Hearst, 1997) and C99 (Choi, 2000) to determine thematic segments boundaries of the text.

The results obtained with C99 were inconclusive: too often, the algorithm cut the text in only two or three thematic segments, even for long texts of about 1500 words. This is clearly insufficient since our goal was to eliminate as many words as possible during the first filtering step.

Even though TextTiling works at the paragraph level, it produces many more thematic segments: it's not rare to get a segment for every paragraph. Despite the fact that it produces so many thematic segments, we decided to use it since we wanted to eliminate as many words as possible via segment filtering. The multitude of segments allows us to retrieve the information necessary to produce the automatic summary.

2.2.3 Sentence segmentation

Since we chose the TextTiling algorithm for thematic segmentation, we then have to segment the paragraphs into sentences. This is done using regular expressions to identify ends of sentences (punctuation) without *overcutting* them such as after a period at the end of an abbreviation.

2.2.4 Basic Elements

We decompose sentences of the corpus into basic elements using the same technique used for question analysis (Section 2.1.2).

2.2.5 Named Entities

Named entities are an important source of information for the comparison of questions to sentences. We detect the named entities and categorize them into one of the four categories introduced in the 2.1.1 section using a Perl package (Cozens and Simoes, 2004) based on a series of regular expressions. For example, a person's name can be detected using prefixes `Mr, Mrs, Dr`, etc. or a company's name can be detected using suffixes `inc, corp, org`, etc. This package also uses a list of proper names of people, companies, places, etc. Moreover, it tries to use the context of the previous sentences to improve the named entity detection of the sentence in question.

So, for every sentence, we keep a list of categorized named entities. For example, consider this sentence:

$$\underbrace{\texttt{Jacques Attali}}_{person}\texttt{, president of the } \underbrace{\texttt{European Bank for Reconstruction and Development}}_{organization}\texttt{, said he would sue for libel over accusations of plagiarism and inaccuracy in his new book about } \underbrace{\texttt{President Mitterand}}_{person}\texttt{.}$$

2.3 Sentence scoring

Scoring and selection of sentences are at the heart of our automatic summarization system.

Before calculating the sentence scores, we filter at the level of thematic segments. We use the cosine similarity measure with an empirically found threshold to identify interesting thematic segments. The calculation is based on the title of the question and the thematic segment. We chose the title since it represents, in most cases, the general topic of the request.

We attribute a score to every sentence in the corpus associated with the question. This score is a linear combination of the following seven measures, all normalized to a value between 0 and 1:

Basic elements We compare the basic elements of the sentences in the question with those of the sentences in the corpus. We compute a score based on the similarity of the words in the constituents that make up the basic elements of the two sentences.

Cosine similarity We compute the cosine similarity measure directly on the sentences in the question and the sentences in the corpus.

Weight of the sentence The sum of the weights of its words obtained based on the IDF file.

Absolute position Score based on the position in the text.

Relative position Score based on the position in the paragraph.

Named Entities We use the results of sections 2.1.1 and 2.2.5 to compute a score for the named entities aspect. We count the number of named entities in the sentence of the corpus that has the same category as a named entity in the question.

Prototypical Expressions We compute the number of prototypical expressions in the sentences, which indicates to us the sentences that have a higher probability of containing important information (for example concluding sentences). We increment a counter for every prototypical expression found in the sentence.

Finally, sentences are sorted in decreasing order of score. 2ex

Lack of redundant information is an important feature of a good summary. Hence, we eliminate sentences that contain too much information similar to that in other sentences. To do this, we use again the cosine similarity measure to decide if two sentences are similar based on an empirically determined threshold. In addition, we compare the named entities in the two sentences, which is a good indicator of whether the sentences talk about the same thing. If they contain two or more common named entities, we consider the sentences as similar and discard the one with the lower score.

Although our system does retrieve the right sentences from the documents to answer a given question, much can be improved. First of all, to improve the selection of sentences, it would be advantageous to identify the keywords of the question, as well as of each thematic segment in the documents, and to compare the two lists. We also tried to find these keywords by comparing the basic elements constituent by constituent, retrieving similar elements. Two elements were considered similar if all their constituents were similar (identical or with a common synonym). We found that this technique retrieved either not enough or too many elements based on the thematic segments, and did not pursue further.

The following sections describe in more detail the measures that we used.

2.3.1 Basic Elements

The basic elements(Hovy et al., 2005) allow us to see the relationship between words inside a sentence. We compare the question to every sentence of the corpus to identify similar relations. We use direct comparison, comparison with the "head" and "mod" inverted, and comparison of synonyms using WordNet (Fellbaum, 1998).

2.3.2 Cosine Similarity

Cosine similarity (Salton, 1989) allows us to compute the similarity of two vectors, which are in our case units of text:

$$Sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} \cdot d_{ij}}{\sqrt{\sum_{j=1}^{t}(d_{ij})^2 \cdot \sum_{j=1}^{t}(w_{qj})^2}}$$

where
$Sim(Q, D_i)$ = similarity between the question Q and the document D_i
d_{ij} = weight of the word T_j in the document D_i
w_{qj} = weight of the word T_j in the question Q

The similarity is measured at the level of words; two units of text would have the highest level of similarity if the words with high weights in one unit also have high weights in the other unit.

We use the $TF \cdot IDF$ measure (Term Frequency, Inverse Document Frequency) to obtain the weights of these words. IDF was computed on the TREC 8, 9, 10 and 11 CD document collection. Stop-words are removed using a stop-words list, and the remaining ones are lemmatized using the Porter algorithm (Porter, 1980).

2.3.3 Sentence weighing

The weight of a sentence is simply the sum of weights of individual words in it:

$$\sum_{i=1}^{n} TF_{w_i} \cdot IDF_{w_i}$$

The summaries should not contain sentences that are meaningless or are not concise, and this measure allows us to identify those sentences that contain the most information, regardless of the question topic.

2.3.4 Sentence position

Relative position The relative position of the sentence is important in the newspaper context: the starting sentences usually describe the topic of the articles, whereas the ending ones often make a condensed summary and/or conclusion.

Absolute position For the absolute position, we used the following formula (Saggion, 2002):

$$\text{absolute score} = \frac{1}{\text{absolute position}}$$

The absolute position of a sentence is its position in the text, from 1 (first sentence) to n (last sentence). The formula favors sentences from the introduction, which we also favor since they contain more information on the topic of the text.

2.3.5 Named Entities

A score is computed based on the number of named entities of the same categories as those detected in the question. We used the following formula:

$$\sum Freq_{Qi} \cdot Freq_{Si}$$

where
$Freq_{Qi}$ is the frequency of words representing the category i in the question
$Freq_{Si}$ is the frequency of NE of category i in the sentence
$i \in$ {*Person, Location, Organization, Time*}

2.3.6 Prototypical Expressions

Certain expressions can reveal in what part of text they appear. We can use these expressions to choose one sentence over another. In our case, we prefer sentences from the introduction or the conclusion, since usually these parts of the text contain more condensed information.

For this measure, we add the number of prototypical expressions (e.g. `as a consequence, as a corollary, as a logical conclusion, as a matter of fact, as a result, as against, as evidence, as far as`) in a sentence without taking into account the type of the expressions (possible types being: pertaining to introduction, conclusion, etc.).

2.4 Post-processing

In order to obtain a more concise and coherent summary, certain operations were done on the sentences to eliminate less important parts or replace certain expressions by other, more concise ones. The following sections describe our processing of the selected sentences.

2.4.1 Temporal expressions resolution

In the course of our experiments, we found that many selected sentences contained temporal expressions that were meaningless taken out of the original context of the article. For example, here's a sentence selected for the question about plagiarism:

```
A former editor for the Wall Street
Journal sued the paper Tuesday for
$12.64 million, claiming that he was
fired and his reputation smeared by a
false charge of plagiarism.
```

In a summary, `Tuesday` doesn't mean anything since the reader doesn't know the date of the article that contained this sentence. We use the module TempEx. The resolution of these expressions takes place in two stages:

1. The temporal expressions are detected in the original documents by the TempEx module.
2. Once the sentences have been chosen, we search the corresponding sentences in the temporary documents. We replace the relevant TimeML markers by the appropriate date format and discard the other ones. Some expressions, such as `several days`, `weekly` or simply `1995` don't need to be replaced.

For the sentence in our previous example, we obtain the following result after the first stage:

```
A former editor for the Wall Street
Journal sued the paper <TIMEX2
TYPE="DATE"
VALUE="19900522">Tuesday</TIMEX2> for
$ 12.64 million , claiming that he
was fired and his reputation smeared
by a false charge of plagiarism .
```

We then use the marker `<TIMEX2>` to find the absolute value for `Tuesday`. Here, it's `19900522` which we transform into the `MM/DD/YY` format and substitute `Tuesday` by this value.

In some cases, we also add a prefix `on` to retain the grammaticality of the sentence. After the second stage, the sentence becomes:

```
A former editor for the Wall Street
Journal sued the paper on 05/22/90
for $12.64 million, claiming that he
was fired and his reputation smeared
by a false charge of plagiarism.
```

2.4.2 Sentence compression

DUC 2005 required the summaries to be at most 250 words long while including as much information as possible. So it is important to eliminate useless pieces of information from the selected sentences in order to incorporate more different sentences into the summary.

First of all, we systematically eliminate text between `()`, `[]`, {}, `--` and -.

For the summaries that ask for general granularity, we use the Collins parser (Collins, 1999) in collaboration with the TreeTagger (Schmid, 1994) part-of-speech tagger. We go through the obtained syntactic tree and eliminate all branches that correspond to subordinate clauses (SBAR) that start with the words `who`, `when`, `where` and `which`.

This operation allows us to remove descriptions of people, places, etc. that are often not necessary for summaries of general granularity.

After these compression operations, it is necessary to repair the sentences. The parser removes the punctuation, which makes the text much more difficult to read. Also, since we remove certain parts of the sentence, it's possible that what is left doesn't end with a period. Hence, we make sure that every sentence ends with a punctuation sign or a closing quote, and that there aren't redundant spaces before it.

2.5 Sentence selection

At the end, our algorithm repeatedly chooses sentences with the highest score from those we selected, until the summary contains at most 250 words. After that, we sort the sentences by date, in increasing order.

3 Evaluation

We now analyze the results of CATS, one of the 32 participants at the DUC 2005 competition.

NIST evaluates summaries in three stages:

Quality A manual evaluation of 5 aspects of linguistic quality:

- **Grammar** The text should not contain non-textual items (i.e. markers) or punctuation errors or incorrect words.
- **Redundancy** The text should not contain redundant information.
- **Clarity of references** The nouns and pronouns should be clearly referred in the summary. For example, the pronoun `he` has to mean something in the context of the summary.
- **Focus** The information presented in the summary must be directly relevant to the topic of the question.
- **Coherence and structure** The summary should have good structure and the sentences should be coherent.

The marks are from A (very good) to E (very poor).

Relevance Does the summary answer the question well, for the chosen granularity? This evaluation is based on the quantity of information provided by the summary in reply to the question. The mark is out of 5 (1 being bad, 5 being very good).

Automatic evaluation ROUGE-1.5.5 is used to compare the automatically generated summaries to those produced manually at NIST. Only the recalls for ROUGE-2 and ROUGE-SU4 are used in the official scoring.

	1	2	3	4	5	Mean
CATS	3.96	4.58	3.46	3.22	2.30	3.50
Best system	4.06	4.48	4.16	3.92	3.22	3.97
Systems' Mean	3.78	4.41	3.01	3.12	2.16	3.29
Humans' Mean	4.81	4.90	4.94	4.89	4.77	4.86

Table 2: Results for the five aspects of linguistic quality of summaries (score out of 5, mean over 50 summaries). The best system represents the one with the best overall mean. 1: Grammar, 2: Redundancy, 3: Clarity of references, 4: Focus, 5: Coherence and structure.

	Total
CATS	2.72
Best system	2.78
Systems' Mean	2.39
Humans' Mean	4.67

Table 3: Results for the relevance of the summaries (score out of 5, Mean over 50 summaries).

3.1 Results

3.1.1 Linguistic quality

Table 2 presents the linguistic quality scores of each question. Overall, we placed 7th (not counting the baseline) on this evaluation.

Grammar Most systems seem to do well in terms of grammar (12th place). We could improve CATS in this respect by correcting words in certain sentences that we choose. For example, the first sentence of a text sometimes starts with several words in capital letters, which is inappropriate for a word in the middle of a summary.

Redundancy The (non) redundancy is very good (4th place), even not far from human-produced summaries. The mean over all systems is however quite high, so this isn't a very meaningful result for the evaluation of a system.

Clarity of references The clarity of references is also very good (3rd place). Despite its good placement, the biggest problem with CATS is that it doesn't re-

	Rouge2	RougeSU4
CATS	0.06	0.13
Best system	0.07	0.13
Systems' Mean	0.06	0.11
Humans' Mean	0.10	0.16

Table 4: Results for the automatic evaluation using ROUGE (recall only, mean over 50 summaries).

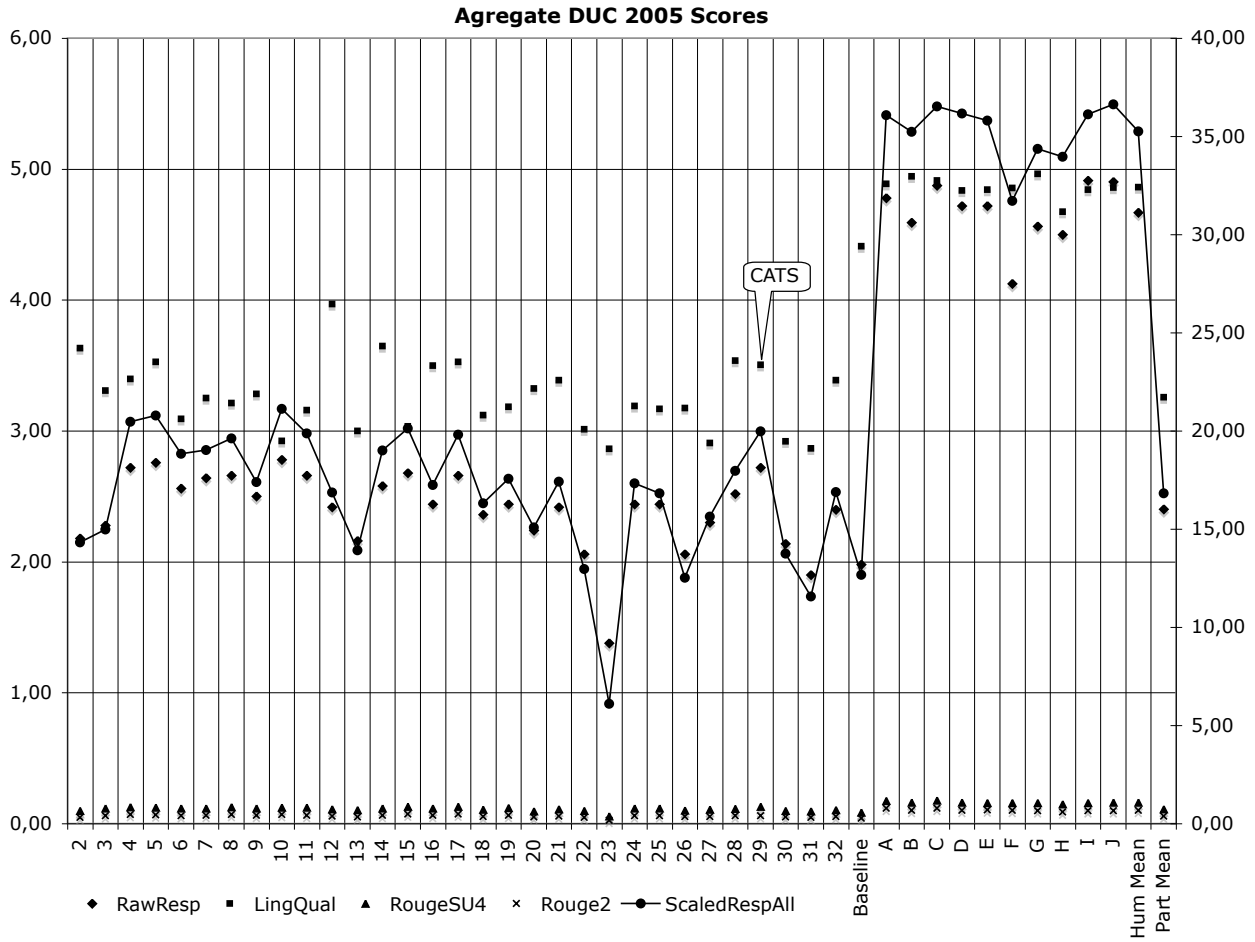

Figure 4: This graph represents the global results for all the systems. The scale on the right is for the *ScaleRespAll* score whose values are on the continuous line The scale on the left is for all other scores. CATS is among the 5 best systems based on all the criteria.

solve personal pronoun references (i.e. `he`, `she`), which causes the same problem as with relative references for dates (section 2.2.1).

Focus Focus could be improved by compressing more the selected sentences. We noticed that the sentences in the corpus are mainly long, so without a good compression algorithm, a lot of unnecessary information makes its way into the summary. CATS got the 11th place.

Coherence and structure The structure is the biggest weakness of CATS which creates the summaries based on a collection of sentences. The structure and the coherence of the resulting sequence of sentences isn't ensured at all but despite, a relatively low score, CATS placed 7th.

3.1.2 Relevance

The relevance of a summary (Table 3) is often considered to be the most important evaluation aspect at this type of conference. CATS did very well (3rd place), barely any worse than the best system. However, there is still a way to go to even approach humans with this respect. The first thing to improve would be the distinction between the two granularities, where the techniques of fusion and sentence generation should be used for the general summaries.

3.1.3 Automatic evaluation

Surprisingly, the automatic evaluation (Table 4) doesn't seem meaningful: the variance of scores is about 0.00008 for ROUGE-2 and about 0.00024 for ROUGE-SU4. Such scores don't let us draw any conclusions.

4 Conclusion

We developed an automatic summarization system that extracts sentences to create 50 summaries of 250 words each, thereby answering 50 complex questions on different topics. We use statistical techniques to compute a score for each sentence in the documents. We then use sentence compression and a cleaning algorithm to shorten the summaries.

To further improve our system, two aspects need to be worked on: sentence compression and the distinction between the two granularities.

To solve our difficulties with sentence compression, it would be interesting to try decomposing complex sentences into multiple simple ones. We could then apply our selection algorithm to these simple phrases instead of the original complex ones, thereby making the summaries more concise.

Acknowledgments

This work was supported by a research grant from the Natural Sciences and Engineering Research Council of Canada.

References

Enrique Amigo, Julio Gonzalo, Victor Peinado, Anselmo Penas, and Felisa Verdejo. 2004. An empirical study of information synthesis tasks. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, Forum Convention Centre Barcelona, July 21-26.

Freddy Y. Y Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of NAACL-00*.

Micheal Collins. 1999. *Head-driven Statistical Models For Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Simon Cozens and Alberto Simoes. 2004. Lingua::EN::NamedEntity.

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

Marti Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Eduard Hovy, Chin-Yew Lin, Liang Zhou, and Junichi Fukumoto. 2005. Basic elements. Technical report, `http://www.isi.edu/~cyl/BE/index.html`.

l. Gerber, L. Ferro, I. Mani, B. Sundheim, G. Wilson, and R. Kozierok. 2002. Annotating temporal information: from theory to practice. In *Proceedings of the 2002 Conference on Human Language Technology*, pages 226–230, San Diego.

Dekang Lin. 1998. Minipar. `http://www.cs.ualberta.ca/~lindek/minipar.htm`.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137, July.

Horacio Saggion. 2002. Shallow-based robust summarization. In *Workshop on Text Summarization (ATALA)*.

G. Salton, 1989. *Automatic text processing*, chapter 9. Addison-Wesley Longman Publishing Co., Inc.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/DecisionTre%eTagger.html`.