CAST: a Computer-Aided Summarisation Tool

Constantin Orăsan, Ruslan Mitkov and Laura Hasler
Research Group in Computational Linguistics
University of Wolverhampton
{C.Orasan, R.Mitkov, L.Hasler}@wlv.ac.uk

Abstract

In this paper we propose computer-aided summarisation (CAS) as an alternative approach to automatic summarisation, and present an ongoing project which aims to develop a CAS system. The need for such an alternative approach is justified by the relatively poor performance of fully automatic methods used in summarisation. Our system combines several summarisation methods, allowing the user of the system to interact with their parameters and output in order to improve the quality of the produced summary.

1 Introduction

The *information overload* predicted a few years ago has become a reality. Recent reports showed that approximately 24 terabytes of text are recorded each year (Lyman and Varian, 2000).[1] Given these values, it becomes increasingly difficult to keep up with the news or find specific information which was produced in the past. Computers play an important role in sifting through information by performing tasks such as classification, retrieval and summarisation.

Automatic summarisation systems help us to deal with the information overload by reducing it. At present the most common type of summarised information is textual information, but unfortunately the quality of the automatic summaries is not of a very high level.

[1]This figure includes books, newspapers, scholarly journals, office documents, etc.

In light of this problem, we propose computer-aided summarisation (CAS) as an alternative to automatic summarisation (AS). Whereas AS does not require any human input to produce summaries, we argue that CAS is a more feasible approach as it allows the user to post-edit the automatic summaries according to their requirements. In this paper we present an ongoing project which in the process of developing CAS environment. The structure of the paper is as follows: In Section 2 we outline related work. Section 3 discusses the objectives of our research, followed by the features of a CAS prototype in the next section. A discussion of current findings and future plans are presented in Section 5, and the paper finishes with concluding remarks.

2 Related work

Apart from a working paper of ours in the mid 90s (Mitkov, 1995), the only relevant research we could find in this field is that of Craven (Craven, 1996). However, Craven's approach takes a rather simplistic view because it uses only methods which extract keywords from the text and not complete sentences or even phrases.

Another tool which aids humans in producing summaries is presented in (Narita, 2000). This tool does not employ any automatic methods to help humans, but gives them the option to access a corpus of human produced abstracts which can function as templates, providing grammatical patterns and collocations common to abstracts.

Endres-Niggemeyer (Endres-Niggemeyer, 1998) identifies three stages in human summarisation: *document exploration*, *relevance assessment* and *summary production*. In the first

two stages the summariser identifies the overall structure of the text and the main topics, whereas in the third one, copy and paste operations followed by post-editing are used to produce the summary. On the basis of these findings, we hypothesise that it is possible to help the summariser by employing well established methods used in automatic summarisation to identify the document's structure, its topic and the most important sentences in the document. After these sentences are identified, the user only needs to improve the quality of the summary.

3 Objectives of the research

The main purpose of our project is to investigate to what extent an automatic summarisation system can help a human summariser produce high quality summaries. A tool which integrates several well-known summarisation methods and which allows a user to run them, combining, filtering and post-editing their results, is being developed. All these operations are completed with the help of a user-friendly interface.

In addition to the tool, this project will give more insights into the summarisation process and the resources needed to produce high quality summaries. It proposes to assess the influence of each of the modules on the final outcome. Given that different methods are embedded in the tool, it will be possible to evaluate and compare them in a common environment.

An important by-product of the project is a corpus of scientific and newspaper articles annotated for summarisation. This corpus contains more information than most corpora of its type, also indicating which clauses from important sentences can be removed without losing information and sentences which need to be extracted so the important sentences can be understood. A description of the corpus can be found in (Hasler et al., 2003).

Simple knowledge-poor summarisation methods prove useful, but are often not good enough to produce high quality summaries. We are currently developing summarisation methods which use the structure of the discourse to determine the best set of sentences to be extracted.

4 The tool

In this section, we briefly present the features of the most important outcome of this project - the computer-aided summarisation tool (CAST).

The tool selects and presents a set of important sentences to the user who can transform the extract into an abstract . As not all sentences identified automatically are worth including in a summary, the user has the option to override the program's decisions and extract additional sentences.

In addition to being used as a computer-aided summarisation tool, CAST can be used as an annotation tool. In this case, different summarisation methods can be combined to highlight important sentences in the text. These sentences can then be saved as gold standard.

The tool can also be used to teach students about summarisation methods. As the tool incorporates several methods, they can be run on the same text, making it possible to compare results. All these methods are highly customisable and the tool enables us to see the influence of different parameters on them.

As aforementioned, the tool relies on several automatic methods to identify the important sentences. At present, these methods are:

Keyword method: Uses TF-IDF scores to weight sentences as proposed in (Zechner, 1996). The user can modify the list of terms and indicate thresholds for sentences' score in order to be considered important

Indicating phrases: Paice (1981) noticed that it is possible to identify phrases which can be used to assess the importance of a sentence. The list of indicating phrases can be loaded, saved and modified in the tool.

Surface clues: Several factors such as sentence location and length can be taken into consideration to decide the importance of the sentence.

Lexical cohesion: Lexical cohesion as proposed in (Hoey, 1991) is used to produce extracts.

Discourse information: Our own summarisation method uses information provided by Centering Theory (Grosz et al., 1995) to produce extracts.

The automatic methods are used not only to identify important sentences, but also to remove sentences which do not contain important

information. For example, it is possible to remove sentences which contain certain indicating phrases or have a TF-IDF score lower than a given threshold. As in the case of important sentences, the user can review the system's decisions.

In order to offer maximum portability CAST, is written in Java, its input being XML. We decided not to include any preprocessing module in CAST (e.g. sentence splitter, PoS tagger, etc.), so all the necessary information needs to be provided in the input file. The advantages of such approach are two-fold: i) it enables people to use their own preprocessing tools; ii) it allows us to test the influence of different preprocessing tools on the results of automatic summarisation methods.

The parameters of all the methods can be adjusted by the user to obtain maximum performance. Their results can be viewed as an extract, or highlighted in the main document via user-defined styles. Given the friendly graphical interface available to the user and the different styles which can be defined by them, different types of sentences can be quickly identified in the text. A screenshot of the tool is presented in Figure 1. A demo of the tool's prototype is also available.

5 Discussion

Section 3 highlighted the main objectives of our research. In this section we explain how these objectives can be achieved.

CAST is intended to help human summarisers to produce abstracts. We conjecture that the summaries produced with the help of the tool will be as good as those manually produced, but it will take less time to write them. In order to prove this hypothesis several experiments will be conducted. In the first one, the time necessary to produce summaries with and without the tool will be recorded. Documents will be summarised manually or with the help of the tool in a random order. After a period of at least 6 months has lapsed so that any effect of familiarity is extinguished, subjects will be asked to produce the same summaries again. However, this time the summaries that have produced done manually, will be done with the aid of the tool, whereas those that have been produced with the help of the tool, will be done manually. The sample of summaries selected for this experiment will be large enough to allow the application of statistical tests.

In a second experiment we envisage to ask humans to decide if a summary was produced with CAST or manually.[2] Our hypothesis is that there will be no significant difference between the two types of summaries and the humans will not be able to make a reliable distinction.

Professional summarisers can summarise the same information with greater competence, speed and quality than non-professionals (Endres-Niggemeyer, 1998). For this reason we intend to run experiments with professional and non-professional summarisers. In this way we hope to show that CAST can be useful for a wide range of users, including professional summarisers.

The user's actions are logged, so it is possible to find out which methods have been used, and with which parameters. Analysis of these logs will give us insight into the way the tool is used, and which of the automatic methods are most useful.

As previously stated, one of the goals of the project is to evaluate each of the summarisation methods integrated in the tool. In order to achieve this, we implemented two evaluation methods. The first one allows a comparison of the output of a summarisation method with a gold standard which is specified by the user. The quality of the summary is computed in terms of precision and recall. The drawback of such a method is that a gold standard is necessary for the evaluation. In order to solve this problem we implemented the second method: a content-based evaluation method which computes the similarity between the summary and its source document, as proposed in (Donaway et al., 2000). The advantage of this method is that it is completely automatic. In addition to evaluating the individual modules, the overall system will also be evaluated.

6 Concluding remarks

Automatic summarisation is still far from delivering high quality results and different

[2] We should emphasise that the comparison will not be between automatic extracts and human produced summaries. Normally, a human does not have problems to make such a distinction. The comparison will be between extracts post-edited by humans with the help of CAST and manually produced summaries.

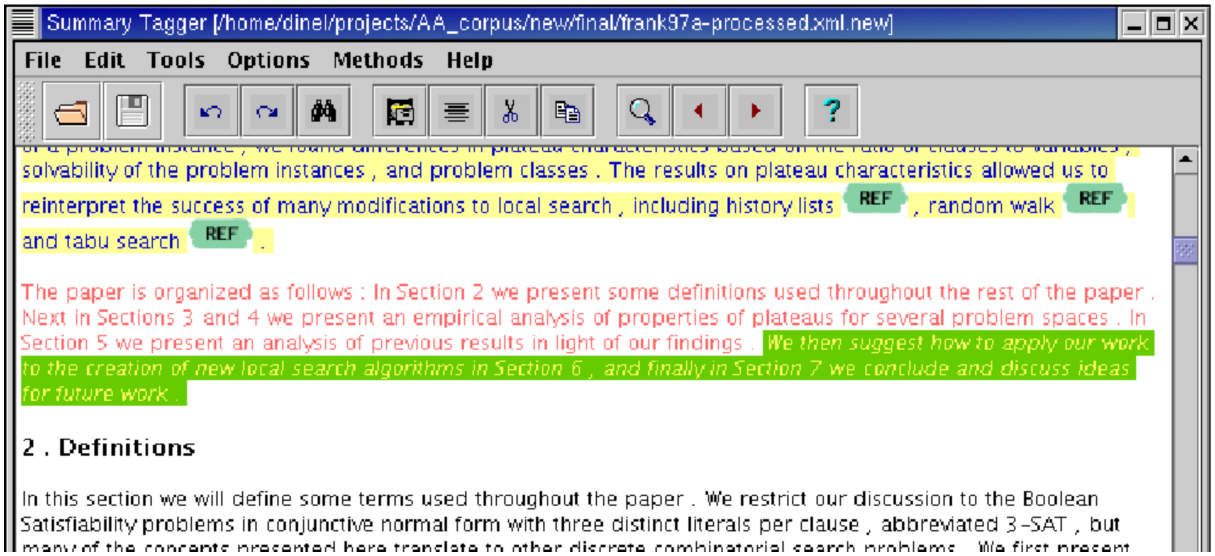

Figure 1: Part of the main screen of the tool

approaches should be considered instead. This paper outlines the development of a computer-aided summarisation tool which promises to be a practical working alternative.

7 Acknowledgements

This work is part of the Arts and Humanities Research Board supported project “A computer-aided summarisation tool”.

References

Timothy C. Craven. 1996. An experiment in the use of tools for computer-assisted abstracting. In *Proceedings of the ASIS 1996*, Baltimore, MD, United States, 19 - 24 October.

Robert L. Donaway, Kevin W. Drummey, and Laura A. Mather. 2000. A comparison of rankings produced by summarization evaluation measures. In *Proceedings of NAACL-ANLP 2000 Workshop on Text Summarisation*, pages 69 – 78, Seattle, Washington, April 30.

B. Endres-Niggemeyer. 1998. *Summarizing information*. Springer.

Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. 1995. Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203 – 225.

Laura Hasler, Constantin Orăsan, and Ruslan Mitkov. 2003. Building beter corpora for summarisation. In *Proceedings of the Corpus Linguistics Conference*, Lancaster, UK, 28th – 31th March.

Michael Hoey. 1991. *Patterns of Lexis in Text*. Describing English Language. Oxford University Press.

Peter Lyman and Hal R. Varian. 2000. How much information. Technical report, School of Information Management and Systems, University of California at Berkeley.

Ruslan Mitkov. 1995. A breakthrough in automatic abstracting: the corpus-based approach. Technical report, University of Wolverhampton.

Masumi Narita. 2000. Constructing a tagged E-J parallel corpus for assisting Japanese software engineers in writing English abstracts. In *Proceedings of the Second International Conference on Language Resources and Evaluation*, pages 1187 – 1191, Athens, Greece, 31 May – 2 June.

Chris D. Paice. 1981. The automatic generation of literature abstracts: an approach based on the identification of self-indicating phrases. In R. N. Oddy, C. J. Rijsbergen, and P. W. Williams, editors, *Information Retrieval Research*, pages 172 – 191. London: Butterworths.

Klaus Zechner. 1996. Fast generation of abstracts from general domain text corpora by extracting relevant sentences. In *COLING - 96, The International Conference on Computational Linguistics*, volume 1, pages 986–989, Center for Sprogteknologi, Copenhagen, Denmark, August.