

Computer-aided summarisation – what the user really wants

Constantin Orăsan and Laura Hasler

Research Group in Computational Linguistics
School of Humanities, Languages and Social Sciences
University of Wolverhampton
Stafford St., Wolverhampton, WV1 1SB
{C.Orasan, L.Hasler}@wlv.ac.uk

Abstract

Computer-aided summarisation is a technology developed at the University of Wolverhampton as a complement to automatic summarisation, to produce high quality summaries with less effort. To achieve this, a user-friendly environment which incorporates several well-known summarisation methods has been developed. This paper presents the main features of the computer-aided summarisation environment and explains the changes introduced to it as a result of user feedback.

1. Introduction

Automatic summarisation systems help us to deal with the information overload by reducing it, but their relatively low quality and domain dependence still makes it impossible to produce human-like summaries from any text. For the last 40 years there has been extensive work in the field of automatic summarisation in an attempt to produce high quality summaries (Luhn, 1958; Edmundson, 1969; Paice, 1981; DeJong, 1982; Brandow et al., 1995; Kupiec et al., 1995; Marcu, 1997; Teufel and Moens, 1997; Mani et al., 1999; Moens, 2000). Some of the proposed methods can be applied to any domain, with the drawback that the summaries produced by them are low quality (Luhn, 1958; Kupiec et al., 1995; Zechner, 1996), whereas others can produce high quality summaries, but only in restricted domains (DeJong, 1982). As a result of this, high quality informative summaries still need to be produced by humans, making the process expensive.

In light of this problem, we propose computer-aided summarisation (CAS) as a complementary approach to automatic summarisation (AS) and a solution to producing high quality summaries at lower costs. Whereas automatic summarisation does not require any human input to produce summaries, we argue that computer-aided summarisation is a more feasible approach as it allows the user to post-edit the automatic summaries according to their requirements, resulting in better finished products.

Computer-aided summarisation is a technology developed at the University of Wolverhampton designed to help humans produce high quality summaries with less effort. This paper presents the computer-aided summarisation technology together with enhancements made to our computer-aided summarisation tool (CAST) as a result of the feedback we received from users. It should be pointed out this paper does not try to prove the usefulness of the underlying concept of computer-aided summarisation. Instead, the paper focuses on usability criteria of the interface, conveying users' views about actually using the computer-aided summarisation tool.

The structure of the paper is as follows: The paper starts with a description of the computer-aided summarisation concept. Section 3. describes the computer-aided

summarisation tool we developed, whilst Section 4. describes the changes we made to the tool as a result of the users' feedback. The paper finishes with a review of related work in the field of computer-aided language processing, followed by conclusions.

2. The computer-aided summarisation concept

It often happens that people are required to summarise documents, but do not have the time or necessary skills to produce high quality summaries. To this end, computer-aided summarisation technology offers a solution by integrating well-known summarisation methods with a user-friendly interface. The concept of computer-aided summarisation was inspired by the *machine-aided translation* approach proposed in 1980 by Martin Kay (Kay, 1980) who defines machine-aided translation as “a cooperative man-machine translation system, leaving the ‘mechanical and routine’ translation work to the computer and ‘the more rewarding, more exciting’ activities to the human translator”. In a similar manner, computer-aided summarisation tries to help the human summariser by selecting the important information from a document. In this way, human effort and time is reduced to linking the extracted sentences in a coherent way and, possibly, removing the redundant information or adding missing information. The main advantage of such an approach is that humans do not need to read the whole text, instead being presented with only the important parts of the document, which then can be edited to suit their needs. However, wherever the information presented seems incomplete, the user has the option to go back to the source and investigate this information in the context in which it originally appears.

The feasibility of the computer-aided summarisation approach is confirmed by research into how humans produce summaries. 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, then in the third stage, copy and paste operations followed by post-editing are used

to produce the summary. Jing and McKeown (Jing and McKeown, 2000) analyse human produced summaries and notice that a large number of *cut and paste* operations are performed on a source document in order to produce the corresponding summary.

On the basis of these findings, we developed a tool that replaces the first two stages identified by Endres-Niggemeyer with automatic summarisation methods which identify the most important sentences in the text. The third step is much more difficult, involving cutting the text and rearranging it in to suitable summary material and therefore cannot be reliably done by the computer. This third stage corresponds to our user editing the important information identified in the first two stages to create a coherent and useful summary. The next section describes the main features of the computer-aided summarisation tool (CAST).

3. The computer-aided summarisation tool

As mentioned above, computer-aided summarisation is seen here as a complement to existing automatic summarisation techniques, as it allows human intervention in the summarisation process. However, in order to make the approach worthwhile, this intervention should be minimal, so that the effort required for a human to produce the summary using CAS is significantly less than that required to write a summary without the help of any tool. In order to achieve this, a wide range of automatic summarisation techniques which have been extensively used in automatic summarisation have been implemented in our computer-aided summarisation tool. The purpose of these methods within CAST is to present to the user an extract which contains the most important sentences from a text, allowing them to post-edit it in order to improve its quality. As not all the sentences identified automatically will be worth including in a summary, the user has the option to override the program's decisions and extract additional sentences, as well as to delete irrelevant sentences.

After careful consideration of the existing automatic summarisation methods commonly used to produce extracts, we decided to implement the following: term specificity weighting methods, methods based on indicating phrases, surface clues, and discourse information. A description of these methods can be found in (Orăsan et al., 2003). Given that each of these methods depend on a host of parameters, we offer users a high level of flexibility without compromising the simplicity of the tool by giving them the option to adjust all these parameters in a very user friendly way.

The automatic methods embedded in the tool are used not only to identify important sentences in a text, but also to remove sentences which do not contain important information. For example, as well as extracting sentences containing certain indicating phrases or having their TF*IDF score above a certain threshold, it is also possible to remove sentences which contain certain indicating phrases or have a TF*IDF score lower than a given threshold. As with the case of important sentences, the user can review the system's decisions overriding it whenever the decision is wrong. The main justification for removing

sentences from the document is that in this way the length of the document to be summarised is reduced, making it easier for the user to quickly browse the document and produce a summary.

The results of the summarisation methods can be viewed in different ways, depending on the user's preferences. They can be viewed either in isolation, when the results are presented as an automatic extract, or the sentences extracted can be highlighted within the source text using styles defined by the user. The advantage of highlighting the results in the text is that the user can easily see the extracted sentences in their original context. Given the friendly graphical interface available to the user and the different styles which can be defined for each method, the user can quickly identify sentences selected by different methods in the text. A screenshot of the tool is presented in Figure 1.

As can be seen in the figure, the interface was designed to be as user-friendly as possible and is split in two windows. The top window contains the text to be summarised and the highlighted output of different automatic summarisation methods. The bottom window is the summary window, into which the user can copy sentences from the full text and edit the summary.

The automatic summarisation methods are used to indicate to the user sentences which are potentially useful. Once a user decides that a sentence is important enough to be included in a summary, it can be copied into the summary window and edited. In order to facilitate the editing task even further, a common set of errors such as dangling pronouns and phrases which could indicate a problem with the summary (e.g. "on the other hand", "secondly", etc.) are highlighted to draw attention to them. Given that compression rate is usually very important when writing a summary, it is continuously updated during the editing process.

4. Feedback from the users

Given such a plethora of useful features, it was believed that users would find the tool very beneficial. To our surprise, they required several additional functionalities to be implemented before they felt completely comfortable using the tool in the summarisation process. These functionalities were mainly related to the user interface as opposed to the underlying computer-aided summarisation methodology. With regard to the automatic summarisation methods, the users preferred to employ only some of them, and in a specific order. First, they ran the term-based summarisation method to identify a set of sentences which contained important information, and then used the most informative sentences as a seed for the lexical chain method in order to find related sentences. In general, the users preferred to mark more sentences than really necessary for the actual summary because this meant that they could first see a wider selection of important sentences and then choose the information they required from these. The main justification for not using all the available methods together was that the users found it slightly confusing to have all the information on the screen at once, especially as the different information is colour-coded.

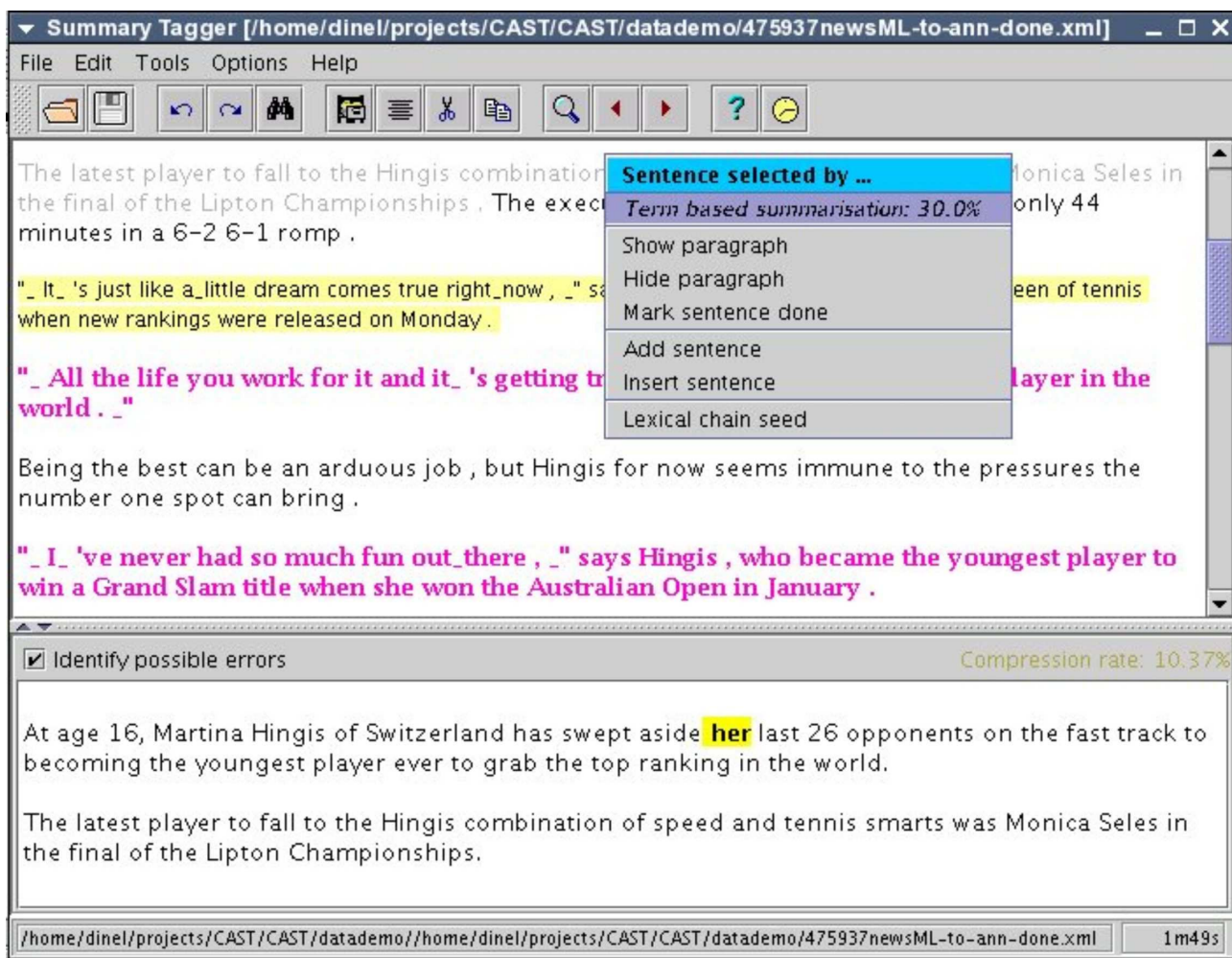


Figure 1: Screenshot of the program

When the user decides that a sentence is important, it can be copied into the summary window located at the bottom of the screen using a simple click. One feature which was required by users was to grey out the copied sentences, as this makes it easier to see straight away that a sentence has been considered for inclusion in the summary. Because a summary is not always produced on the basis of full sentences, the users also wanted a quick way to insert a sequence of words as opposed to full sentences in the summary window.

Other features required by the users were undo/redo operations and a simpler way to identify sentences selected by more than one method. The facility to run automatic summarisation methods in a different window from which it is possible to copy sentences into the summary window was also implemented. The justification for this feature is that in this way it is possible to increase the length of the automatic extract more quickly, without affecting the text in the main window, so that the user can easily see if any additional important information can be identified and transfer this information if necessary.

5. Related work in computer-aided summarisation

Apart from a working paper in the mid 90s (Mitkov, 1995), the only relevant research we could find in the field of computer-aided summarisation is that of Craven (Craven, 1996). However, Craven's approach differs from ours in that it takes a rather simplistic view using only methods which extract keywords and not complete sentences from the text. The experiments reported by Craven indicate that even a tool which relies on such a simple method can be beneficial for summarisers, as over 34% of the subjects found the lists of keywords provided very useful or quite useful. However, the paper concludes that the accessibility of the tool needs to be improved in order to obtain better results.

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. Such an approach proved very useful for researchers writing abstracts in non-native languages.

The sparseness of research in computer-aided

summarisation is rather surprising given that similar approaches proved very useful in other fields: computer-aided translation was proved to facilitate the work of translators (Mitkov, 1994), the time necessary to generate multiple-choice questions reduced to a quarter when a computer-aided approach was used (Mitkov and Ha, 2003) and it is common to use semi-automatic methods to speed up the production of annotated corpora.

6. Conclusions

In this paper, we presented the concept of computer-aided summarisation with the emphasis on users' requirements for such a tool. The underlying hypothesis of computer-aided summarisation is that it is possible to speed up the summary writing process by employing automatic summarisation techniques to produce a basic summary and then allowing the user to edit the text as necessary. However, the feedback we received from the users indicates that whilst these techniques are important for producing a basis for the user summary, they are not everything. Our major finding is the extent to which a user-friendly interface that integrates these techniques can be important.

7. Acknowledgments

This research was partially funded by the Arts and Humanities Research Board through the "Computer Aided Summarisation Tool - CAST" project.¹

8. References

- Ronald Brandow, Karl Mitze, and Lisa F. Rau. 1995. Automatic condensation of electronic publications by sentence selection. *Information Processing & Management*, 31(5):675 – 685.
- 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.
- G. DeJong. 1982. An overview of the FRUMP system. In W. G. Lehnert and M. H. Ringle, editors, *Strategies for natural language processing*, pages 149 – 176. Hillsdale, NJ: Lawrence Erlbaum.
- H. P. Edmundson. 1969. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264 – 285, April.
- B. Endres-Niggemeyer. 1998. *Summarizing information*. Springer.
- Hongyan Jing and Kathleen McKeown. 2000. Cut and paste based text summarization. In *Proceedings of the 1st Conference of the North American Chapter of the Association for Computational Linguistics (NAACL'00)*, pages 178 – 185, Seattle, Washington, May.
- Martin Kay. 1980. The proper place of men and machines in language translation. Technical Report CSL-80-11, Xerox PARC, Palo Alto, California.
- Julian Kupiec, Jan Pederson, and Francine Chen. 1995. A trainable document summarizer. In *Proceedings of the 18th ACM/SIGIR Annual Conference on Research and Development in Information Retrieval*, pages 68 – 73, Seattle.
- H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159 – 165.
- Inderjeet Mani, Barbara Gates, and Eric Bloedorn. 1999. Improving summaries by revising them. In *Proceedings of the 37th Annual Meeting of the ACL*, pages 558 – 565, University of Maryland, College Park, Maryland, USA, 20 – 26 June.
- Daniel Marcu. 1997. *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*. Ph.D. thesis, Department of Computer Science, University of Toronto, Toronto, Canada.
- Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17 – 22, Edmonton, Canada, May.
- Ruslan Mitkov. 1994. Intelligent machine-aided translation: a realistic alternative. In *Proceedings of the 1994 ACLIC/PacFoCoL Conference*, Kyoto, Japan.
- Ruslan Mitkov. 1995. A breakthrough in automatic abstracting: the corpus-based approach. Technical report, University of Wolverhampton.
- Marie-Francine Moens. 2000. *Automatic Indexing and Abstracting of Document Texts*. Kluwer Academic Publishers.
- 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.
- Constantin Orăsan, Ruslan Mitkov, and Laura Hasler. 2003. CAST: a Computer-Aided Summarisation Tool. In *Proceedings of EACL2003*, pages 135 – 138, Budapest, Hungary, April.
- 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.
- Simone Teufel and Marc Moens. 1997. Sentence extraction as a classification task. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scallable Text Summarization*, pages 58 – 59, Madrid, Spain, July 11.
- 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.

¹The project's webpage is <http://clg.wlv.ac.uk/projects/CAST/>