An evolutionary approach for improving the quality of automatic summaries

Constantin Orăsan
Research Group in Computational Linguistics
School of Humanities, Languages and Social Sciences
University of Wolverhampton
C.Orasan@wlv.ac.uk

Abstract

Automatic text extraction techniques have proved robust, but very often their summaries are not coherent. In this paper, we propose a new extraction method which uses local coherence as a means to improve the overall quality of automatic summaries. Two algorithms for sentence selection are proposed and evaluated on scientific documents. Evaluation showed that the method ameliorates the quality of summaries, noticeable improvements being obtained for longer summaries produced by an algorithm which selects sentences using an evolutionary algorithm.

1 Introduction

It is generally accepted that there are two main approaches for producing automatic summaries. The first one is called *extract and rearrange* because it extracts the most important sentences from a text and tries to arrange them in a coherent way. These methods were introduced in the late 50s (Luhn, 1958) and similar methods are still widely used.

The second approach attempts to understand the text and, then, generates its abstract, for this reason it is referred to as *understand and generate*. The best-known method that uses such an approach is described in (DeJong, 1982). Given that the methods which "understand" a text are domain dependent, whenever robust methods are required, extraction methods are preferred.

Even though the extraction methods currently used are more advanced than the one proposed in (Luhn, 1958), many still produce summaries which are not very coherent, making their reading difficult. This paper presents a novel summarisation approach which tries to improve the quality of the produced summaries by ameliorating their local cohesion.

This paper is structured as follows: In Section 2 we present our hypothesis: it is possible to produce better summaries by enforcing the *continuity principle* (see next section for a definition of this principle) . A corpus of scientific abstracts is analysed in Section 3 to learn whether this principle holds in human produced summaries. In Section 4, we present two algorithms which combine traditional techniques with information provided by the *continuity principle*. Several criteria are used to evaluate these algorithms on scientific articles in Section 5. We finish with concluding remarks, which also indicate possible future research avenues.

2 How to ensure local cohesion

In the previous section we already mentioned that we are trying to improve the automatic summaries by using the *continuity principle* defined in Centering Theory (CT) (Grosz et al., 1995). This principle, requires that two consecutive utterances have at least one entity in common. Even though it sounds very simple, this principle is important for the rest of the principles defined in the CT because if it does not hold, none of the other principles can be satisfied. Given that only the *continuity principle* will be used in this paper and due to space limits, the rest of these principles are not discussed here. Their description can be found in (Kibble and Power, 2000). For the same reason we will not go into details about the CT.

In this paper, we take an approach similar to (Karamanis and Manurung, 2002) and try to produce summaries which do not violate the *continuity principle*. In this way, we hope to produce summaries which contain sequences of sentences that refer the same entity, and therefore will be more coherent. Before we can test if the principle is satisfied, it is necessary to define certain parameters on which the principle relies. As aforementioned, the principle is tested on pairs of consecutive utterances. In general utterances are clauses or sentences. Given that the automatic identification of clauses is not very accurate, we consider sentences as utterances. An advantage of using sentences is that most summarisation methods extract sentences, which makes it easier to integrate them with our method.

In this paper, we consider that two utterances have an entity in common if the same head noun phrase appears in both utterances. In order to determine the head of noun phrases we use the FDG tagger (Tapanainen and Järvinen, 1997) which also provides partial dependency relations between the constituents of a sentence. At this stage we do not employ any other method to determine whether two noun phrases are semantically related.

3 Corpus investigation

Before we implemented our method, we wanted to learn if the *continuity principle* holds in human produced summaries. In order to perform this analysis we investigated a corpus of 146 human produced abstracts from the *Journal of Artificial Intelligence Research* (JAIR). [1]

Most of the processing was done automatically using a simple script which tests if the principle is satisfied by pairs of consecutive utterances (i.e. if the pair has at least one head noun phrase in common). Those pairs which violate the principle were manually analysed.

In our corpus almost 75% of the pairs of consecutive utterances (614 out of 835) satisfy the principle. In terms of summaries, it was noticed that 44 out of 146 do not have any such pairs which violate the principle.

After analysing the violations, we can explain them in one of the following ways:

– In 126 out of 221 cases (57%) the link between utterances is realised by devices such as rhetorical relations.

– In 76 cases (34%) the *continuity principle* was realised, but was not identified by the script because of words were replaced by semantic equivalents. In only 17 of these cases pronouns were used.

– Ramifications in the discourse structure violate the principle in 19 cases (9%). These ramifications are usually explicitly marked by phrases such as *firstly, secondly*.

After investigating our corpus we can definitely say that the *continuity principle* is present in human produced abstracts, and therefore by trying to enforce it in automatic summaries, we might produce better summaries. However, by using such approach we cannot be sure that the produced summaries are coherent, being known that it is possible to produce cohesive texts, but which are incoherent. In Section 4 we present a method which uses the *continuity principle* to score the sentences. This method is then evaluated in Section 5.

We also have to emphasise that we do not claim that humans consciously apply the *continuity principle* when they produce summaries or any other texts. The presence of the violations identified in our corpus is an indication for this.

4 The method

Karamanis and Manurung (2002) used the *continuity principle* in text generation to choose the most coherent text from several produced by their generation system. In their case, the candidate texts were sequences of facts, their best ordering was determined by an evolutionary algorithm which tried to minimise the number of violations of the *continuity principle* they contained.

We take a similar approach in our attempt to produce coherent summaries, trying to minimise the number of violations of the principle they contain. However, our situation is more difficult because

[1] The full articles and their abstracts are freely available at http://www.jair.org

a summarisation program needs firstly to identify the important information in the document and then present it in a coherent way, whereas in text generation the information to be presented is already known. "Understand and generate" methods would be appropriate, but they can only be applied to restricted domains. Instead, we employ a method which scores a sentence not only using its content, but also considering the context in which the sentence would appear in a summary. Two different algorithms are proposed. Both algorithms use the same content-based scoring method (see Section 4.1), but they use different approaches to extract sentences. As a result, the way the context-based scoring method defined in Section 4.2 is applied differs. The first algorithm is a greedy algorithm which does not always produce the best summary, but it is simple and fast. The second algorithm employs an evolutionary technique to determine the best set of sentences to be extracted.

We should point out that another difference between our method and the ones used in text generation is that we do not intend to change the order of the extracted sentences. Such an addition would be interesting, but preliminary experiments did not lead to any promising results.

4.1 Content-based scoring method

We rely on several existing scoring methods to determine the importance of a sentence on the basis of its content. In this section we briefly describe how this score is computed. The heuristics employed to compute the score are:

Keyword method: uses the TF-IDF scores of words to compute the importance of sentences. The score of a sentence is the sum of words' scores from that sentence (Zechner, 1996)

Indicator phrase method: Paice (1981) noticed that in scientific papers it is possible to identify phrases such as *in this paper, we present, in conclusion*, which are usually meta-discourse markers. A list of such phrases has been built and all the sentences which contain an indicating phrase have their scores boosted or penalised depending on the phrase.

Location method: In scientific papers important sentences tend to appear at the beginning and end of the document. For this reason sentences in the first and the last 13 paragraphs have their scores boosted. This value was determined through experiments.

Title and headers method: Words in the title and headers are usually important, so sentences containing these words have their scores boosted.

Special formatting rules: Quite often certain important or unimportant information is marked in texts in a special way. In scientific paper it is common to find equations, but they rarely appear in the abstracts. For this reason sentences that contain equations are excluded.

The score of a sentence is a weighted function of these parameters, the weights being established through experiments. As already remarked by other researchers, one of the most important heuristics proved to be the indicating phrase method.

4.2 Context-based scoring method

Depending on the context in which a sentence appears in a summary, its score can be boosted or penalised. If the sentence which is considered satisfies the *continuity principle* with either the sentence that precedes or follows it in the summary to be produced, its score is boosted.[2] If the *continuity principle* is violated the score is penalised. After experimenting with different values we decided to boost the sentence's score with the TF-IDF scores of the common NPs' heads and penalise with the highest TF-IDF score in the document.

While analysing our corpus we noticed that large number of violations of the *continuity principle* are due to utterances in different segments. Usually this is explicitly marked by a phrase. We extracted a list of such phrases from our corpus and decided not to penalise those sentences which violate the *continuity principle*, but contain one of these phrases.

4.3 The greedy algorithm

The first of the two sentence selection algorithms is a greedy algorithm which always extracts the highest scored sentence from those not extracted yet. The sentences' scores are computed in the way described

[2] The way the sentences which precedes and follows it is determined depends very much on the algorithm used (see Sections 4.3 and 4.4 for details). If the sentence is the first or the last in a summary (i.e. there is no preceding or following sentence) the score is not changed.

	Given an extract $\{S_{summ_1}, S_{summ_2},..., S_{summ_m}\}$ and S the sentence which is considered for extraction
1.	Find S_{prec} and S_{next} from the extract which are the closest sentences before and after S in the document, respectively.
2.	Adjust the score S considering the context S_{prec}, S, S_{next}.

Figure 1: The way the weights of a sentence are adjusted by the greedy algorithm

in Section 4.2. Given that the original order of sentences is maintained in the summary, whenever a sentence is considered for extraction, the algorithm presented in Figure 1 is used. We should emphasise that at this stage the sentence is not extracted, but its score is computed as if it is included in the extract. After this process is completed for all the sentences which are not present in the extract, the one with the highest score is extracted. The process is repeated until the required length of the summary is reached. As it can be noticed, the algorithm cannot be applied to the first sentence. For this reason the first extracted sentence is always the one with the highest content-based score.

It should be noted that it is possible to extract a sentence S_2 which satisfies the continuity principle with its preceding sentence S_1, but in a later iteration to extract another sentence, which is between these two, and which satisfies the continuity principle with S_1, but not with S_2. Unfortunately, given the nature of the algorithm, it is impossible to go back and replace S_2 with another sentence, and therefore sometimes the algorithm does not find the best set of sentences. In order to alleviate this problem, in the next section we present an algorithm which selects sentences using an evolutionary algorithm.

4.4 The evolutionary algorithm

The greedy algorithm presented in the previous section selects sentences in an interactive manner, the inclusion of a sentence in the summary depending on the sentences which were included before. As a result it is possible that the best summary is not produced. In order to alleviate this problem an algorithm which uses evolutionary techniques to select the set of sentences is proposed.

Evolutionary algorithms are advanced searching algorithms which use techniques inspired by the nature to find the solution of a problem. A specific type of evolutionary algorithms are genetic algorithms (Holland, 1975) which encode the problem as a series of genes, called *chromosome*. The most common way to encode genes is the binary encoding, where each gene can take the values 0 or 1. If we have decided to use such an encoding the value 0 would have meant not to include the sentence in the summary, whereas 1 to include it. For our problem the length of a chromosome would have been equal to the number of sentences in the texts. For long texts, such as the ones we use, this would have meant very long chromosomes, and as a result slow convergence, without any certainty that the best solution is found (Holland, 1975).

3	5	8	10	14	18	66	79

Figure 2: A chromosome representing a summary which contains the sentences 3, 5, 8, 10, 14, 18, 66, 79 from the document

Instead of using binary encoding, we decided that our genes take integer values, each value representing the position of a sentence from the original document to be included in the summary. The length of the chromosome is the desired length of the summary. Caution needs to be taken whenever a new chromosome is produced so the values of the genes are distinct (i.e. the summary contains distinct sentences). If a duplication is found in a chromosome, then the gene's value which contains the duplication is incremented by one. In this way the chromosome will contain two consecutive sentences, and therefore it could be more coherent. A chromosome is presented in Figure 2.

Genetic algorithms use a fitness function to assess how good a chromosome is. In our case the fitness function is the sum of the scores of the sentences indicated in the chromosome. The sentences' scores are not considered "in isolation", they are adjusted in the way described in Section 4.2. For this algorithm, determining the preceding and the following sentence is trivial, all the information being encoded in the chromosome.

Genetic algorithms use genetic operators to evolve a population of chromosomes (Holland, 1975). In our case, we used *weighed roulette wheel selection* to select chromosomes. Once several chromosomes are selected they are evolved using *crossover* and *mutation*. We used the classical *single point crossover* operator and two mutation operators. The first one replaces the value of a gene with a randomly generated integer value. The purpose of this operator is to try to include random sentences in the summary and in this way to help the evolutionary process. The second mutation operator replaces the values of a gene with the value of the preceding gene incremented by one. This operator introduces consecutive sentences in the summary, which could improve coherence.

The genetic algorithm starts with a population of randomly generated chromosomes which is then evolved using the operators. Each of the operators has a certain probability of being applied. The best chromosome (i.e. the one with the highest fitness score) produced during all generations is the solution to our problem. In our case we iterated a population of 500 chromosomes for 100 generations. Given that the search space (i.e. the set of sentences from the document) is very large we noticed that at least 50 generations are necessary until the best solution is achieved. The algorithm is evaluated in the next section.

5 Evaluation and discussion

We evaluated our methods on 10 scientific papers from the *Journal of Artificial Intelligence Research*, totalising almost 90,000 words. The number of texts used for evaluation might seem small, but given that from each text we produced eight different summaries which had to be read and assessed by humans, the evaluation process was very time consuming.

Throughout the paper we have mentioned the term *quality of a summary* several times without defining it. In this paper the quality of a summary is measured in terms of *coherence*, *cohesion* and *informativeness*. The coherence and cohesion were quantified through direct evaluation using a methodology similar to the one proposed in (Minel et al., 1997). The cohesion of a summary is indicated by the number of *dangling anaphoric expressions*,[3] whereas the coherence by the *number of ruptures in the discourse*. For informativeness we computed the similarity between the automatic summary and the document as proposed in (Donaway et al., 2000). Given that the methods discussed in this paper try to enforce local coherence they directly influence only the number of discourse ruptures, the changes of the other two measures are a secondary effect.

In our evaluation, we compared the two new algorithms with a baseline method and the content-based method. The baseline, referred to as TF-IDF, extracts the sentences with the highest TF-IDF scores. The comparison with the baseline does not tell us if by adding the context information described in Section 4.2 the quality of a summary improves. In order to learn this, we compared the new algorithms with the one presented in Section 4.1. They all use the same content-based scoring method, so if differences were noticed, they were due to the context information added and the way sentences are extracted.

The results of the evaluation are presented in Tables 1, 2 and 3. In these tables *TF-IDF* represents the baseline, *Basic method* is the method described in section 4.1, whereas *Greedy* and *Evolutionary* are the two algorithms which use the *continuity principle*. In Table 1, the row *Maximum* indicates the maximum number of ruptures which could be found in that summary. This number is given by the total number of sentences in the summary.

Given that for the direct evaluation the summaries had to be analysed manually, in a first step, we produced 3% summaries. After noticing only slight improvement when using our methods, we decided to increase their lengths to 5%, to learn if the methods perform better when they produce longer summaries. The values for the 5% summaries are represented in the tables in brackets.

[3] A *dangling anaphor* is a referential expression which is deprived of its referent as a result of extracting only the sentence with the anaphoric expression.

Method	Text										Total
	1	2	3	4	5	6	7	8	9	10	
TFIDF	12 (29)	5 (13)	17 (33)	10 (16)	7 (10)	12 (19)	9 (15)	14 (18)	12 (35)	8 (15)	106 (203)
Basic method	8 (24)	4 (11)	11 (23)	5 (7)	4 (6)	7 (14)	9 (8)	12 (11)	10 (16)	7 (12)	77 (132)
Greedy	8 (20)	4 (7)	12 (20)	4 (10)	4 (7)	8 (16)	11 (7)	8 (9)	9 (14)	8 (12)	76 (122)
Evolutionary	6 (11)	3 (9)	14 (16)	4 (5)	4 (4)	7 (9)	7 (3)	8 (3)	9 (9)	5 (6)	67 (75)
Maximum	15 (39)	12 (21)	20 (51)	13 (20)	7 (13)	15 (23)	14 (23)	15 (25)	17 (44)	11 (40)	139 (299)

Table 1: The number of discourse ruptures in the summaries

5.1 Number of ruptures in the discourse

A factor which reduces the legibility is the number of discourse ruptures (DR). Using an approach similar to (Minel et al., 1997) we consider that a discourse rupture occurs when a sentence seems completely isolated from the rest of the text. Usually this happens due to presence of isolated discourse markers such as *firstly, secondly, however, on the other hand, etc.* Table 1 shows the number of DR in these summaries.

A result which was expected is the large number of DR in the summaries produced by our baseline. Such a result is normal given that the method does not use any kind of discourse information. The baseline is outperformed by the rest of the methods in almost all the cases, the overall number of DR for each method being significantly lower than the DR of the baseline.

Table 1 shows that for 3% summaries, the context information has little influence on the number of the discourse ruptures present in a summary. This suggests that the information provided by the indicating phrases (which are meta-discourse markers) has greater influence on the coherence of the summary than the *continuity principle*.

The situation changes when longer summaries are considered. As can be observed in Table 1, the *continuity principle* reduces the number of DR; this number for the *Evolutionary algorithm* being almost half the number for *Basic method*. Actually, by examining the table, we can see that the evolutionary algorithm performs better than the basic method in all of the cases. The same cannot be said about the greedy algorithm. It performs more or less the same as the basic algorithm, the overall improvement being negligible. This clearly indicates that in our case a simple greedy algorithm is not enough to choose the set of sentences to extract, and more advanced techniques need to be used instead.

The methods proposed in this paper perform better when longer summaries are produced. Such a result is not obtained only because the summary contains more sentences, and is therefore more likely to contain sentences which are related to each other. If this was the case, we would not have such a large number of DR in summaries generated by the baseline. We believe that the improvement is due to the discourse information used by the methods.

If the values of DR for each text are scrutinised, we can notice very mixed values. For some of the texts the continuity principle helps a lot, but for others it has little influence. This suggests that for some of the texts the continuity principle is too weak to influence the quality of a summary, and a combination of the continuity principle with the other principles from centering theory, as already used for text generation in (Kibble and Power, 2000), could lead to better summaries.

The methods proposed in this paper rely on several parameters to boost or penalise the scores of a sentence on the basis of context. A way to improve the results of these methods could be by selecting better values for these parameters.

5.2 Dangling anaphors

Even though the problem of anaphora is not directly addressed by our methods, a subsidiary effect of the improvement of the local cohesion should be a decrease in the number of dangling references.

Table 2 contains the number of dangling references in the summaries produced by different

Method	Text										Total
	1	2	3	4	5	6	7	8	9	10	
TFIDF	12 (31)	3 (25)	22 (35)	13 (15)	4 (10)	14 (22)	14 (16)	11 (22)	12 (19)	9 (15)	144 (210)
Basic method	12 (26)	2 (23)	17 (29)	7 (13)	2 (7)	11 (20)	10 (9)	10 (8)	6 (12)	8 (15)	85 (162)
Greedy	11 (19)	3 (14)	15 (20)	4 (19)	3 (9)	13 (23)	16 (10)	4 (11)	7 (12)	7 (14)	83 (151)
Evolutionary	8 (18)	3 (16)	15 (18)	6 (6)	2 (6)	9 (12)	10 (7)	4 (5)	5 (13)	7 (12)	69 (113)

Table 2: Number of dangling anaphors in the summaries

methods. This number reduces in the summaries produced by the evolutionary algorithm. As in the case of discourse ruptures, the greedy algorithm does not perform significantly better than the basic method. All the methods outperform the baseline.

We noticed that the most frequent dangling references were due to phrases referring to tables, figures, definitions and theorems (e.g. *As we showed in Table 3 ...*). They can be referred to in any point in the text, and therefore, the local coherence cannot guarantee inclusion of the referred entities. Moreover, in many cases the referred entity is not necessarily textual (e.g. tables and figure), and therefore should not be included in a summary. In light of these, we believe that the problem of such dangling references should be addressed by the content-based method, which normally should filter sentences containing them.

Dangling referential pronouns are virtually nonexistent, which means that in most of the cases the reader can understand, at least partially, the meaning of the referential expression.

As observed for DR, the values for individual texts are mixed.

5.3 Text informativeness

In order to assess whether information is lost when the context-based method is used to enhance the sentence selection, we used a content-based evaluation metric (Donaway et al., 2000). This metric computes the similarity between the summary and the whole document, a good summary being one which has a value close to 1.[4]

Table 3 shows that the evolutionary algorithm does not lead to major loss of information, for several text this method obtains the highest score. In contrast, the greedy method seems to exclude useful information, for several texts, performing worse than the basic method and the baseline.

[4]In this paper we used cosine distance between the document's vector and the automatic summary's vector. Before building the vectors the texts were lemmatised.

6 Related work

In text summarisation several researchers have addressed the problem of producing coherent summaries. In general, rules are applied to revise summaries produced by a summarisation system (Mani et al., 1999; Otterbacher et al., 2002). These rules are produced by humans who read the automatic summaries and identify coherence problems. Marcu (2000) produced coherent summaries using Rhetorical Structure Theory (RST). A combination of RST and lexical chains is employed in (Alonso i Alemany and Fuentes Fort, 2003) for the same purpose. Comparison to the work by Marcu and Alonso i Alemany is difficult to make because they worked with different types of texts. As already mentioned, information from centering theory was used in text generation to select the most coherent text from several candidates (Kibble and Power, 2000; Karamanis and Manurung, 2002).

7 Conclusion

In this paper we presented two algorithms which combine content information with context information. The first one is a greedy algorithm which chooses one sentence at a time, but once a sentence is selected it cannot be discarded. The second algorithm employs an evolutionary technique to determine the set of extracted sentences, overcoming the limitations of the first algorithm.

Evaluation on scientific articles showed that the

Method	Text									
	1	2	3	4	5	6	7	8	9	10
TF-IDF	0.84 (0.92)	0.85 (0.95)	0.84 (0.93)	0.92 (0.87)	0.87 (0.94)	0.80 (0.90)	0.86 (0.87)	0.92 (0.86)	0.82 (0.89)	0.88 (0.85)
Basic method	0.81 (0.91)	0.85 (0.87)	0.87 (0.90)	0.93 (0.87)	0.89 (0.93)	0.88 (0.87)	0.89 (0.83)	0.90 (0.89)	0.68 (0.88)	0.92 (0.86)
Greedy	0.87 (0.90)	0.85 (0.94)	0.80 (0.89)	0.93 (0.88)	0.86 (0.95)	0.84 (0.74)	0.78 (0.85)	0.90 (0.86)	0.58 (0.84)	0.90 (0.88)
Evolutionary	0.82 (0.86)	0.88 (0.95)	0.84 (0.91)	0.94 (0.89)	0.86 (0.88)	0.87 (0.88)	0.90 (0.88)	0.86 (0.87)	0.81 (0.82)	0.88 (0.91)

Table 3: The similarity between the summary and the document from which it is produced

evolutionary method performs consistently better than the rest of the methods in terms of coherence and the cohesion, and does not degrade the information content in most of the cases.

From each text we produced 3% and 5% summaries. For the 3% summaries there is no significant improvement when contextual information is used (not even when the evolutionary algorithm is used). However, for 5% summaries, the number of discourse ruptures in the summaries produced by the evolutionary algorithm is almost half the number of DR in the ones produced by the basic method. The number of dangling referential expressions also reduces. Regardless the length of the summary, it seems to be no significant difference between the basic method and the greedy algorithm.

One could argue that for long documents, 5% summaries are too long, and that shorter versions are required. This is true, but these summaries can be shortened by using aggregation rules like the ones proposed in (Otterbacher et al., 2002), where two sentences referring to the same entity are merged into one. Given that the summaries produced with the evolutionary algorithm contain more sequences of sentences related to the same entity, it will be easier to apply such aggregation rules.

As noted in Section 5.1, the results vary from one text to another. In some cases the *continuity principle* noticeably improves the quality of a summary, but in other cases the improvement is moderate or low. One reason could be that the *continuity principle* alone is too weak to be able to guarantee the coherence of the produced summary. We intend to extend our experiments and test whether a combination of centering theory's principles, as used in (Kibble and Power, 2000), can lead to better results.

Our algorithms were tested on scientific articles. We intend to extend the evaluation using other types of texts in order to learn if the genre influences the results.

To conclude, in this paper we argue that it is possible to improve the quality of automatic summaries by using the *continuity principle* and by employing an evolutionary algorithm to select sentences. This improvement seems to be text dependent, in some cases being small.

Acknowledgements

Preparation of this paper was supported by the Arts and Humanities Research Board through the CAST project.

References

Laura Alonso i Alemany and Maria Fuentes Fort. 2003. Integrating cohesion and coherence for automatic summarisation. In *Proceedings of EACL2003*, pages 1 – 8, Budapest, Hungary, April.

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.

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.

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.

J.H. Holland. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Nikiforos Karamanis and Hisar Maruli Manurung. 2002. Stochastic text structuring using the principle of continuity. In *Proceedings of International Natural Language Generation Conference*, pages 81 – 88, New York, USA, 1 – 3 July.

Rodger Kibble and Richard Power. 2000. An integrated framework for text planning and pronominalisation. In *Proceedings of International Natural Language Generation Conference*, pages 77 – 84, Mitzpe Ramon, Israel, 12 - 16 June.

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. 2000. *The theory and practice of discourse parsing and summarisation*. The MIT Press.

J Minel, S Nugier, and G Piat. 1997. How to appreciate the quality of automatic text summarization? In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scallable Text Summarization*, pages 25 – 30, Madrid, Spain, July 11.

Jahna C. Otterbacher, Dragomir R. Radev, and Airong Luo. 2002. Revisions that improve cohesion in multi-document summaries: A preliminary study. In *Proceedings of the Workshop on Text Summarization*, pages 27 – 36, University of Pennsylvania, Philadelphia, PA, USA, 11 – 12 July.

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.

P. Tapanainen and T. Järvinen. 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing*, pages 64 – 71, Washington D.C., USA.

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.