KNOWLEDGE ENGINEERING: PRINCIPLES AND TECHNIQUES
Proceedings of the International Conference on Knowledge Engineering,
Principles and Techniques, KEPT2007
Cluj-Napoca (Romania), June 6–8, 2007, pp. 17–24

THE ROLE OF LINGUISTIC INFORMATION FOR SHALLOW LANGUAGE PROCESSING

CONSTANTIN ORĂŞAN[1]

Abstract. Many methods in computational linguistics rely on shallow processing to achieve their goals. The advantage of these methods in comparison to deep processing is that they do not require the building of elaborate representations of the text to be processed or to perform reasoning on this data, and as a result they can be more easily implemented. This paper shows how shallow methods for automatic summarisation can improve their performance by adding different types of linguistic information.

1. Introduction

In language processing it is generally accepted that two approaches can be employed. On the one hand there are *deep linguistic approaches* which build an elaborate representation of the problem they resolve in order to "understand" the texts they process and make inferences. On the other hand there are *shallow linguistic approaches* where different types of information are extracted from the text and then combined in order to solve the problem tackled, but no attempt is made to understand the text they process. Deep linguistic approaches have been widely used to implement different grammatical formalism, but they were also used with various degrees of success in real-world applications such as information extraction and automatic summarisation. The drawback of these methods is that they lack robustness and coverage due to the fact that quite often they rely on hand coded resources. In contrast, shallow approaches are robust at the expense of performance, which usually is lower than that of deep processing. Due to the fact that they require less effort to implement they have been widely used for all kind of purposes. In this paper, we show that it is possible to overcome some of the limitations of shallow processing methods and improve their performance by combining different linguistic information. In order to prove this, automatic summarisation is taken as a case study. The paper is structured as follows: Section 2 briefly presents existing methods in automatic summarisation classifying them in

2000 *Mathematics Subject Classification.* 91F20, 68T35.
Key words and phrases. automatic summarisation, evaluation, shallow processing.

shallow and deep methods. Section 3 presents and evaluates different automatic summarisation methods showing how the results improve with the addition of linguistic information. Section 4 discusses the results and concludes the paper.

2. Deep vs. shallow processing in automatic summarisation

The field of *automatic summarisation* develops automatic methods which try to replace the human summarisers by producing summaries using automatic means. Unfortunately, with the current technology it is difficult to produce automatic summaries which are of similar quality with summaries produced by professional summarisers. Instead, it can produce *indicative summaries* which can indicate the content of a document and can help readers decide whether the content is relevant to their interests.

In general, in automatic summarisation two main approaches are employed: *automatic extraction methods* and *automatic abstraction methods* [10]. The former produce *extracts* which are sets of units (i.e. sentences, paragraphs or clauses) extracted with no or little modifications from the source text(s), and normally employ shallow linguistic processing. The later produce *abstracts* which present the most important information in the text to be summarised, but contain units not present in the source. Quite often, in order to produce abstracts deep linguistic processing is required. The remaining of this section presents a brief overview of the methods employed in automatic summarisation with emphasis on whether they rely on deep or shallow processing.

The way most of the shallow automatic summarisation methods work is to determine a score for each sentence and on the basis of this score, extract the sentences with the highest scores until the desired length of summary is reached. The first such summarisation method relied on the distribution of words to determine sentences which contain important information for a text Luhn [14]. Even though the method was proposed almost 45 years ago, its promising results encouraged other researchers to apply similar approaches, in many cases in combination with other methods [7, 13, 25, 24, 9]

Edmundson [7] noticed that the presence of certain words can indicate that a sentence is important or that it can be discarded during the summarisation process. Given the beneficial influence of this method on the quality of extracts it was extended to phrases [20] , and now is widely used in combination with other methods [13, 24, 11]. In a similar manner *named entities* were used as an indicator of a sentence's importance [13, 22].

Shallow processing was also used to determine the discourse structure of texts and produce summaries. Cue words and phrases were used in [18, 16, 5] to derive the rhetorical structure of a text [15] and employ it in the summarisation process. Links between entities in a text were also used to produce summaries. Boguraev and Kennedy [4] and Azzam et. al. [1] employed anaphoric and coreferential links, whilst Barzilay and Elhadad [2] focused on lexical repetition.

A general characteristic of the methods presented above is that they do not try to "understand" the text. In contrast, methods which rely on deep linguistic processing try to imitate the way humans produce summaries by understanding a text first and then generate an abstract on the basis of information understood. Because, in order to function, these systems require large quantities of information about the domain, these methods are also called *knowledge rich methods*. The downside of this approach is the fact that the systems are domain dependent, which means they cannot easily be ported to different domains or to be used in domain independent contexts.

The best known system based on deep linguistic processing to produce summaries is FRUMP [6]. The approach taken by this system relies on *sketchy scripts* to encode information about the events it can "understand", the participants in these events, and the way in which the participants interact with each other and with the environment. The participants were identified using surface clues, and their actions traced using a simple inference engine. Rumelhart [21] developed a system to understand and summarise simple stories, using a grammar which generated semantic interpretations of the story on the basis of hand-coded rules. The SUSY system [8] is of particular interest because it tries to implement the theory proposed by Kintsch [12] to understand and summarise a text, and therefore attempts to replicate the way humans summarise texts. This system relies on linguistic knowledge to understand the meaning and structure of a text, and on world knowledge to reason and infer new information.

3. Linguistic information for shallow automatic summarisation

The previous section has presented several methods used in automatic summarisation. In this section some of these methods are implemented and evaluated, showing that in many cases addition of linguistic information has a beneficial effect on the informativeness of summaries. This section starts with a description of the corpus used in the experiments and the evaluation method employed. Sections 3.2 - 3.6 present the summarisation methods investigated here, followed by their evaluation in Section 3.7.

3.1. **Corpus and evaluation method.** For the experiments described in this paper, a corpus of journal articles published in the Journal of Artificial Intelligence Research (JAIR) was used. This corpus contains 65 articles with over 600,000 words. In order to assemble this corpus, the electronic versions of these articles were downloaded and converted to plain text. For the purpose of automatic summarisation the corpus was automatically annotated with sentence boundaries, token boundaries and part-of-speech information using the FDG tagger [23]. To evaluate the performance of automatic summarisation methods the author produced abstract was identified and extracted from the article.

The evaluation measure used in this paper calculates the similarity between an automatic extract and the author produced abstract using the *cosine measure*, a very popular measure for determining the similarity between two vectors. The formula to calculate this is:

$$\cos(\vec{S_a}, \vec{S_h}) = \frac{\sum_{i=1}^{n} S_a(i) S_h(i)}{\sqrt{\sum_{i=1}^{n} S_a(i)}\sqrt{\sum_{i=1}^{n} S_h(i)}} \tag{1}$$

where S_a and S_h are the vectors built from the automatic and human summaries respectively, n is the number of distinct words in $S_a \cup S_h$, and $S_a(i)$ and $S_h(i)$ are the frequencies of word i in S_a and S_h respectively. In order to make the similarity value more accurate, a stoplist is applied before building the vectors.

3.2. **Upper limit and baseline.** Using the evaluation method presented in the previous section, it is possible to identify in every text a set of sentences which has the maximum similarity score with the source's human abstract. This maximum figure represents the upper limit any extraction method could reach, and indicates that the only way to further increase similarity is to produce an abstract. In this paper, we employ the method proposed in [19] to identify the upper limit of extraction methods using both the greedy algorithm and the genetic algorithm proposed in that paper, depending on which compression rate is used. The results of the upper limit are presented in Table 1.

A baseline is usually a very simple method which does not really employ much knowledge to produce a summary, and which is normally used for comparison. In this research, it was decided to consider as baseline a method which extracts the first and last sentences from paragraphs, starting with the first paragraph in the text, until the desired length is achieved. The justification for this baseline can be found in research by Baxendale [3] who noticed that the first and last sentences from paragraphs are more important than others in scientific documents. The results of this baseline are presented in Section 3.7.

3.3. **Term-based summarisation.** Term-based summarisation assumes that the importance of a sentence can be determined on the basis of the words it contains. The most common way of achieving this is to weight all the words in a text, and calculate the score of a sentence by adding together the weights of the words within it. In this way, a summary can be produced by extracting the sentences with the highest scores until the desired length is reached. In this section two token weighting methods are used in the summarisation process: term frequency and TF*IDF. Each of them is presented next.

3.3.1. *Term frequency.* It was noticed that when a person writes a text, he or she normally repeats concepts as they progress through the text, and those concepts which are repeated most are the ones which are linked to the main topics of the text [14]. Using this observation, it is possible to assign to each token a score equal to its frequency in order to indicate the topicality of the concept represented by it.

$$Score(t) \,=\, TF_t \,=\, the\, frequency\, of\, token\, \mathbf{t}\, in\, the\, text \tag{2}$$

The main drawback of term frequency is that it wrongly assigns high scores to frequent tokens such as prepositions and articles. For this reason, a stoplist is used to filter out these words.

3.3.2. *TF*IDF.* The elimination of stopwords does not ensure that only important tokens receive a high score. In order to address this problem, *document frequency* can be used. The assumption of this measure is that the importance of a token is inversely proportional to the number of documents in which it appears. The inverse document frequency (IDF) on its own is a relatively weak indicator of the token's importance, and for this reason very often it is used in conjunction with the term frequency. The formula used to calculate TF*IDF is:

$$TF * IDF(t) \,=\, TF_t \,*\, log\,\frac{N}{n_t} \tag{3}$$

where N is the number of documents in the collection, n_t is the number of documents in the collection which contain the token t and TF_t is the term frequency as calculated in the previous section.

3.4. **Anaphora resolution for automatic summarisation.** The term-based summariser presented in the previous section relies on word frequencies to calculate the score of a sentence, but because some of these words are referred to by pronouns the frequency of the concepts they represent are not correctly calculated. In this section, a robust anaphora resolver is used to assign semantic information to pronouns and in this way obtain more reliable frequency counts of concepts, which in turn improves the results of the term-based summariser. After experimenting with several anaphora resolvers, MARS [17], a robust anaphora resolution method which relies on a set of boosting and impeding indicators to select the antecedent of a pronoun, was selected. In order to evaluate the performance of MARS a third of the corpus was annotated with anaphoric links. The results of the evaluation indicate a success rate of around 51%.

3.5. **Indicating phrases.** First introduced by Paice [20], indicating phrases are groups of words which can be used to determine the importance of a sentence that contains them. For scientific domain typical examples of indicating phrases are phrases such as *in this paper we present, we conclude that.* In order to acquire a list of indicating phrases relevant to the scientific domain, a corpus of scientific abstracts was used to extract a list of 4-grams which was then manually edited. Sentence were scored according to how many indicating phrases they contained.

Compression rate	2%	3%	5%	6%	10%
Baseline	0.260	0.327	0.419	0.440	0.479
Term-based summariser using TF	0.415	0.443	0.461	0.468	0.484
Term-based summariser using TF*IDF	0.396	0.427	0.467	0.472	0.496
Summariser which uses indicating phrases	0.428	0.452	0.492	0.500	0.527
Term-based summariser using TF and MARS	0.455	0.480	0.494	0.500	0.513
Term-based summariser using TF*IDF and MARS	0.428	0.463	0.499	0.503	0.520
Combination	0.480	0.496	0.532	0.535	0.541
Upper limit of extraction methods	0.725	0.743	0.753	0.748	0.788

Table 1. The informativeness of summaries produced using different methods

3.6. **Combination of modules.** The term-based summariser and the summariser based on indicating phrases are rarely used independently. For this reason, a summariser which combines information from the two modules was implemented. In this summariser the scores assigned by the term-based summarisation module and the module based on indicating phrases are normalised to a value between 0 and 1, and are combined using a linear function. After experiments, it was decided to give each module a weight of 1.

3.7. **Evaluation results.** In order to evaluate the influence of different linguistic information on the informativeness of summaries produced 2%, 3%, 5%, 6% and 10% summaries have been produced for each text. Table 1 presents the results of the evaluation.

As expected, all the methods investigated in this paper perform better than the baseline. Term-based summarisers produce significantly better results than the baseline for all compression rates but 10% summaries where the differences are not statistically significant. Comparison between the results of the two term-based summarisers reveal that the performance of the two methods depend on the compression rate. For high compression rates (i.e. 2% and 3%) it is necessary to weight words using TF, whereas for lower compression rates TF*IDF should be used. The same phenomenon happens when the term-based summariser is enhanced with information from the anaphora resolver. The summariser based on indicating phrases performs significantly worse than the enhanced term-based summariser at high compression rates, but at lower compression rates (i.e. 5%, 6% and 10%) the differences are not statistically significant, or it performs better than the enhanced summariser. The combination of modules leads to much better results than running each module individually.

4. Discussion and conclusions

In this paper we investigated the influence of shallow linguistic information on the quality of automatic summarisation. To achieve this several automatic summarisation methods were developed and evaluated. First a method to determine the upper limit of extraction methods was run to show that by using shallow linguistic methods, which produce extracts and not abstracts, there is a limit on how similar the automatic summaries can be in comparison to human abstracts. All the methods evaluated in the paper performed well below this upper limit.

The paper has also shown that by combining shallow linguistic information it is possible to improve the informativeness of automatic summaries. A baseline which relies on non-linguistic positional information was outperformed by all the methods which benefit from linguistic information. Simple lexical information about the frequency of tokens was enough to produce better results than the baseline. When more accurate words frequency are calculated using semantic information provided by an anaphora resolver the results improve significantly. Very similar summaries from the point of view of informativeness are produced using indicating phrases which can be considered both lexical information and pseudo-discourse markers. A summariser which combines all this information produces better summaries indicating that combination of shallow linguistic information can lead to better results.

References

[1] Saliha Azzam, Kevin Humphrey, and Robert Gaizauskas. Using coreference chains for text summarisation. In Amit Bagga, Breck Baldwin, and Sara Shelton, editors, *Coreference and Its Applications*, pages 77 – 84, University of Maryland, College Park, Maryland, USA, June 1999.

[2] Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automated Text Summarization*, pages 111 – 121. The MIT Press, 1999.

[3] Phyllis B. Baxendale. Man-made index for technical literature - an experiment. *I.B.M. Journal of Research and Development*, 2(4):354 – 361, 1958.

[4] Branimir Boguraev and Christopher Kennedy. Salience-based content characterisation of text documents. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automated Text Summarization*, pages 99 – 110. The MIT Press, 1999.

[5] Simon H. Corston-Oliver. Beyond string matching and cue phrases: Improving the efficiency and coverage in discourse analysis. In *AAAI Spring Symposium on Intelligent Text Summarisation*, pages 9 – 15, Stanford, California, USA, March 23-25 1998.

[6] G. DeJong. 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, 1982.

[7] H. P. Edmundson. New methods in automatic extracting. *Journal of the Association for Computing Machinery*, 16(2):264 – 285, April 1969.

[8] Danilo Fum, Giovanni Guida, and Carlo Tasso. Evaluating importance: a step towards text summarisation. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pages 840 – 844, Los Altos CA, August 1985.

[9] Le An Ha and Constantin Orăsan. Concept-centred summarisation: producing glossary entries for terms using summarisation methods. In *Proceedings of RANLP2005*, pages 219 – 225, Borovets, Bulgaria, September 21 – 23 2005.

[10] Eduard Hovy. Text summarisation. In Ruslan Mitkov, editor, *The Oxford Handbook of computational linguistics*, pages 583 – 598. Oxford University Press, 2003.

[11] Eduard Hovy and Chin-Yew Lin. Automated text summarization in SUMMARIST. In Inderjeet Mani and Mark T. Maybury, editors, *Advances in Automated Text Summarization*, pages 81 – 94. The MIT Press, 1999.

[12] Walter Kintsch. *The representation of meaning in memory*. The Experimental psychology series. Lawrence Erlbaum Associates Publishers, 1974.

[13] Julian Kupiec, Jan Pederson, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th ACM/SIGIR Annual Conference on Research and Development in Information Retrieval*, pages 68 – 73, Seattle, July 09 – 13 1995.

[14] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159 – 165, 1958.

[15] Willliam C. Mann and Sandra A. Thompson. Rhetorical Structure Theory: Description and construction of text structures. In *NATO Advanced Research Workshop on Natural Language Generation*, pages 85 – 95. 1986.

[16] Daniel Marcu. From discourse structures to text summaries. In Inderjeet Mani and Mark Maybury, editors, *Proceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summarization*, pages 82 – 88, Madrid, Spain, 1997. ACL.

[17] Ruslan Mitkov, Richard Evans, and Constantin Orăsan. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. In *Proceedings of CICLing-2002*, pages 168 – 186, Mexico City, Mexico, February 2002.

[18] Kenji Ono, Kakuo Sumita, and Seiji Miike. Abstract generation based on rhetorical structure extraction. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING-94)*, pages 344 – 348, Kyoto, Japan, 1994.

[19] Constantin Orăsan. Automatic annotation of corpora for text summarisation: A comparative stuty. In *Proceedings of the 6th International Conference CICLing2005*, pages 670 – 681, Mexico City, Mexico, February 2005. Springer-Verlag.

[20] Chris D. Paice. 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, 1981.

[21] E. Rumelhart. Notes on a schema for stories. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding: Studies in Cognitive Science*, pages 211 – 236. Academic Press Inc, 1975.

[22] Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended named entity hierarchy. In *Third International Conference on Language Resources and Evaluation (LREC 2002)*, pages 1818 – 1824, Las Palmas de Gran Canaria, Spain, May 2002.

[23] Pasi Tapanainen and Timo Järvinen. A non-projective dependency parser. In *Proceedings of the 5th Conference of Applied Natural Language Processing*, pages 64 – 71, Washington D.C., USA, March 31 - April 3 1997.

[24] Simone Teufel and Marc Moens. 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 1997.

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

[1]Research Group in Computational Linguistics, University of Wolverhampton, Wolverhampton, UK

E-mail address: C.Orasan@wlv.ac.uk