Top-Down Cohesion Segmentation in Summarization

Doina Tatar
Andreea Diana Mihis
Gabriela Serban
University "Babeş-Bolyai" Cluj-Napoca (Romania)
email: dtatar@cs.ubbcluj.ro

Abstract

The paper proposes a new method of linear text segmentation based on lexical cohesion of a text. Namely, first a single chain of disambiguated words in a text is established, then the rips of this single chain are considered as boundaries for the segments of the cohesion text structure (Cohesion TextTiling or CTT). The summaries of arbitrarily length are obtained by extraction using three different methods applied to the obtained segments. The informativeness of the obtained summaries is compared with the informativeness of the pair summaries of the same length obtained using an earlier method of logical segmentation by text entailment (Logical TextTiling or LTT). Some experiments about CTT and LTT methods are carried out for four "classical" texts in summarization literature showing that the quality of the summarization using cohesion segmentation (CTT) is better than the quality using logical segmentation (LTT).

1 Introduction

Text summarization has become the subject of an intense research in the last years and it is still an emerging field (Orasan, 2006; Radev et al., 2002; Hovy, 2003; Mani, 2001). The research is done in the extracts (which we are treating in this paper) and abstracts areas. The most important task of summarization is to identify the most informative (salient) parts of a text comparatively with the rest. A good segmentation of a text could help in this identification (Boguraev and Neff, 2000; Barzilay and Elhadad, 1999; Reynar, 1998).

This paper proposes a new method of linear text segmentation based on lexical cohesion of a text. Namely, first a single chain of disambiguated words in a text is established, then the rips of this chain are considered. These rips are boundaries of the segments in the cohesion structure of the text. Due to some similarities with TextTiling algorithm for topic shifts detection of Hearst (1997), the method is called Cohesion TextTiling (CTT).

The paper is structured as follows: in Section 2 we present the problem of Word Sense Disambiguation by a chain algorithm and the derived CTT method. In Section 3, some notions about textual entailment and logical segmentation of a text by LTT method are discussed. Summarization by different methods after segmentation is the topic of Section 4. The parallel application of CTT and LTT methods to four "classical" texts in summarization literature, two narrative and two newspapers, and some statistics of the results are presented in Section 5. We finish the article with conclusions and possible further work directions.

2 A top-down cohesion segmentation method

2.1 Lexical chains

A lexical chain is a sequence of words such that the meaning of each word from the sequence can be obtained unambiguously from the meaning of the rest of words (Morris and Hirst, 1991; Barzilay and Elhadad, 1999; Harabagiu and Moldovan, 1997; Silber and McCoy, 2002; Stokes, 2004). The map of all lexical chains of a text provides a representation of the lexical cohesive structure of the text. Usually a lexical chain is obtained in a bottom-up fashion, by taking each candidate word of a text, and finding an appropriate relation offered by a thesaurus as Rodget (Morris and Hirst, 1991) or WordNet (Barzilay and Elhadad, 1999). If it is found, the word is inserted with the appropriate sense in the current chain, and the senses of the other words in the chain are updated. If no relation is found, then a new chain is initiated.

Our method approaches the construction of lexical chains in a reverse order: we first disambiguate the whole text and then construct the lexical chains which cover as much as possible the text.

2.2 CHAD algorithm

It is known that in the last years many researchers studied the possibility to globally disambiguate a text. In Tatar et al. (2007) is presented CHAD algorithm, a Lesk's type algorithm based on WordNet, that doesn't require syntactic analysis and syntactic parsing. As usually for a Lesk's type algorithm, it starts from the idea that a word's dictionary definition is a good indicator for the senses of this word and uses the defi-

nition in the dictionary directly. The base of the algorithm is the disambiguation of a triplet of words, using Dice's overlap or Jaccard's measures. Shortly, CHAD begins with the disambiguation of a triplet $w_1w_2w_3$ and then adds to the right the following words to be disambiguated. Hence it disambiguates at a time a new triplet, where first two words are already associated with the best senses and the disambiguation of the third word depends on the disambiguation of these first two words.

Due to the brevity of definitions in WordNet (WN), the first sense in WN for a word w_i (WN 1^{st} sense) must be associated in some cases in a "forced" way. The **forced condition** represents the situation that any sense of w_i is related with the senses of the words w_{i-2} and w_{i-1}. Thus the **forced condition** signals that a lexical chain stops, and, perhaps, a new one begins.

Comparing the precision obtained with CHAD and the precision obtained by the WN 1^{st} sense algorithm for 10 files of Brown corpus (Tatar et al., 2007) we obtained the result: for 7 files the difference was greater or equal to 0.04 (favorable to WN 1^{st}), and for 3 files was lower. For example, in the worst case (Brown 01 file), the precisions obtained by CHAD are: 0.625 for Dice's measure, 0.627 for Overlap measure, 0.638 for Jaccard's measure while the precision obtained by WN 1^{st} sense is 0.688. Let us remark that CHAD is used to mark the discontinuity in cohesion, while WN 1^{st} sense algorithm is unable to do this.

2.3 CHAD and lexical chains

The CHAD algorithm shows what words in a sentence are unrelated as senses with the previously words: these are the words which receive a "forced" first WN sense. Of course, these are regarded differently from the words which receive a "justified" first WN sense. Scoring each sentence of a text by the number of "forced" to first WN sense words in this sentence, we will provide a representation of the lexical cohesive structure of the text. If F is this number, then the valleys (the local minima) in the graph representing the function $1/F$ will represent the boundaries between lexical chains (see Figure 2).

Lexical chains could serve further as a basis for an algorithm of segmentation. As our method of determination of lexical chains is linear, the corresponding segmentation is also linear. The obtained segments could be used effectively in summarization. In this respect, our method of summarization falls in the discourse-based category. In contrast with other theories about discourse segmentation, as Rhetorical Structure Theory (RST) of Mann and Thompson (1988), attentional/intentional structure of Grosz and Sidner (1986) or parsed RST tree of Marcu (1997), our CTT method (and also, as presented below, our LTT method) supposes a linear segmentation (versus hierarchical segmentation) which results in an advantage from a computational viewpoint.

3 Segmentation by Logical TextTiling

3.1 Text entailment

Text entailment is an autonomous field of Natural Language Processing and it represents the subject of some recent Pascal Challenges. As is established in an earlier paper (Tatar et al., 2007), a text T entails an hypothesis H, denoted by $T \rightarrow H$, iff H is less informative than T. A method to prove $T \rightarrow H$ which relies on this definition

consists in the verification of the relation: $sim(T,H)_T \leq sim(T,H)_H$. Here $sim(T,H)_T$ and $sim(T,H)_H$ are text-to-text similarities introduced in Corley and Mihalcea (2005). The method used by our tool for Text entailment verification calculates the similarity between T and H by *cosine*, thus the above relation becomes $cos(T,H)_T \leq cos(T,H)_H$ (Tatar et al., 2007).

3.2 Logical segmentation

Tatar et al. (2008) present a method named *logical* segmentation because the score of a sentence is the number of sentences of the text which are entailed by it. Representing the scores of sentences as a graph, we obtain a structure which indicates how the most important sentences alternate with ones less important and which organizes the text according to its logical content. Simply, a valley (a local minimum) in the obtained logical structure of the text is a boundary between two logical segments (see Figure 1).

The method is called Logical TextTiling (LTT), due to some similarities with the TextTiling algorithm for topic shifts detection (Hearst, 1997). The drawback of LTT, that the number of the segments is fixed for a given text (as it results from its logical structure), is eliminated by a method to dynamically correlate the number of the logical segments obtained by LTT with the required length of the summary. Let us remark that LTT does not require a predicate-argument analysis. The only semantic structure processing required is the Text Entailment verification.

4 Summarization by segmentation

4.1 Scoring the segments

An algorithm of segmentation has usually the following function:

INPUT: a list of sentences $S_1,...,S_n$ and a list of scores $score(S_1),...,score(S_n)$;

OUTPUT: a list of segments $Seg_1,...,Seg_N$.

Given a set of N segments (obtained by CTT or LTT) we need a criterion to select those sentences from a segment which will be introduced in the summary. Thus, after the score of a sentence is calculated, we calculate a score of a segment. The final score, $Score_{final}$, of a sentence is weighted by the score of the segment which contains it. The summary is generated by selecting from each segment a number of sentences proportional with the score of the segment. The method has some advantages when a desired level of granularity of summarization is imposed.

The summarization algorithm *with Arbitrarily Length of the summary (AL)* is the following:

INPUT: The segments $Seg_1,...Seg_N$, the length of summary X (as parameter), $Score_{final}(S_i)$ for each sentence S_i;

OUTPUT: A summary SUM of length X, where from each segment Seg_j are selected $NSenSeg_j$ sentences. The method of selecting the sentences is given by definitions $Sum1, Sum2, Sum3$ (Section 4.2).

Remark: A number of segments Seg_j **may** have $NSenSeg_j > 1$. If $X < N$ then a number of segments Seg_j **must** have $NSenSeg_j = 0$

In Section 5 (Experiments) the variant of summarization algorithm as above is denoted as **Var1**. In the variant **Var2** a second choice of computing the score of a segment is considered. Namely, the score is not normalized, and it is equal with the sum of its sentences scores, without been divided to the segment length. The drawback of **Var1** is that in some cases a very long segment can contain some sentences with a high score and many sentences with a very low score, the final score of this segment will be a small one and those important sentences will not be included in the final summary. The drawback of **Var2** is that of increased importance of the length of the segment in some cases. Thus, the score of a short segment with high sentences scores will be less then one of a long segment with small sentences scores, and again some important sentences will be lost.

4.2 Strategies for summary calculus

The method of extracting sentences from the segments is decisive for the quality of the summary. The deletion of an arbitrary amount of source material between two sentences which are adjacent in the summary has the potential of losing essential information. We propose and compare some simple strategies for including sentences in the summary:

- Our first strategy is to include in the summary the first sentence from each segment, as this is of special importance for a segment. The corresponding summary will be denoted by Sum_1.
- The second way is that for each segment the sentence(s) with a maximal score are considered the most important for this segment, and hence they are included in the summary. The corresponding summary is denoted by Sum_2.
- The third way of reasoning is that from each segment the most informative sentence(s) (the least similar) relative to the previously selected sentences are picked up. The corresponding summary is denoted by Sum_3.

5 Experiments

In our experiments for CTT method each sentence is scored as following: $Score(S_i) = \frac{1}{nuw_i}$ where nuw_i is the number of words "forced" to get the first WN sense in the sentence S_i. If $nuw_i = 0$ then $Score(S_i) = 2$. The graph of the logical structure for the text **Hirst** is presented in Figure 1 while the graph for the cohesion structure for the same text is presented in Figure 2.

We have applied CTT and LTT methods of segmentation and summarization to four texts denoted in the following by: **Hirst** (Morris and Hirst, 1991), **Koan** (Richie, 1991), **Tucker1** (Tucker, 1999) and **Tucker2** (Tucker, 1999).[1] The denotations are as following: *LSi* for LTT with *Sumi* method, *CSi* for CTT with *Sumi* method. Also an ideal summary (*IdS*) has been constructed by taking the majority occurrences of the sentences in all *LSi* and *CSi* summaries. *IdS* is the last raw of the table. For the text **Tucker1** the summaries with five sentences obtained by us and the summary obtained by the author with CLASP (Tucker, 1999) are presented in Table 1.

[1] All these texts are shown on-line at `http://www.cs.ubbcluj.ro/~dtatar/nlp/` (first entries).

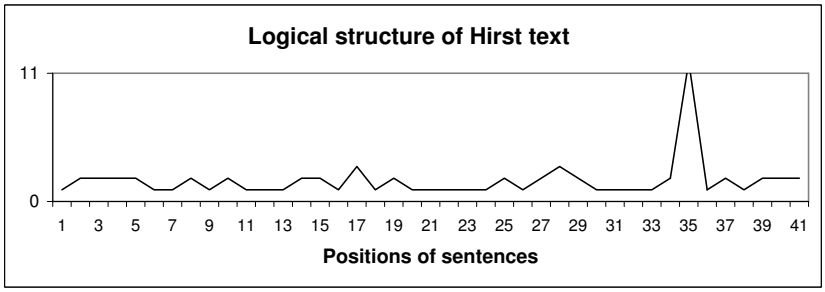

Figure 1: The logical structure of the text **Hirst**

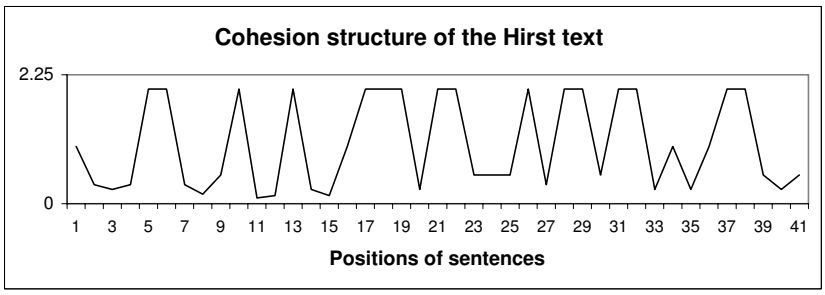

Figure 2: The cohesion structure of the text **Hirst**

Table 1: The summaries with the length 5 for the text **Tucker1** compared with the author's summary

Method	**5 sent**	**Tucker1**
*LS*1	1,16,17,34,35	6,8,16,23,34
*LS*2	1,16,17,34,35	
*LS*3	1,32,34,43,44	
*CS*1	1,23,31,34,40	
*CS*2	8,23,31,35,43	
*CS*3	1,9,27,34,39	
IdS	1,16,17,34,35	

Table 2: The average informativeness of CTT and LTT summaries for all texts

Method	Var1	Var2	Var1 + Var2
*LS*1	0.606538745	0.603946109	0.605242427
*LS*2	0.580429322	0.577647764	0.579038543
*LS*3	0.594914426	0.600104854	0.59750964
*CS*1	0.607369111	0.603053171	0.605211141
*CS*2	0.592993154	0.589675201	0.591334178
*CS*3	0.631625044	0.594702506	0.613163775
average	0.602311633	0.594854934	0.598583284
LTTaverage	0.593960831	0.593899576	0.593930203
CTTaverage	**0.6106624**	**0.5958102**	**0.6032363**

5.1 Evaluation of the summarization

There is no an unique formal method to evaluate the quality of a summary. In this paper we use as a measure of the quality of a summary, the similarity (calculated as *cosine*) between the summarized (initial) text and the summaries obtained with different methods. We call this similarity "the informativeness".

The informativeness of the different types of summaries Sum_1, Sum_2, Sum_3 (see Section 4.2) and of different lengths (5, 6 and 10) is calculated for each text. Then, the average informativeness for all four texts is calculated. A view with the these average results of informativeness, calculated with different methods, in variants **Var1** and **Var2**, is given in the Table 2.

Let us remark that for obtaining summaries with different lengths, after a first segmentation with CTT and LTT methods the algorithm *AL* from Section 4.1 is applied.

Table 2 displays the results announced in the abstract: the quality of CTT summaries is better than the quality of the LTT summaries from the point of view of informativeness.

5.2 Implementation details

The methods presented in this paper are fully implemented: we used our own systems of Text Entailment verification, Word Sense Disambiguation, top-down lexical chains determination, LTT and CTT segmentation, summarization with Sum_i and *AL* methods. The programs are realized in Java and C++. WordNet (Miller, 1995) is used by our system of Word Sense Disambiguation.

6 Conclusion and further work

This paper shows that the text segmentation by lexical chains and by text entailment relation between sentences are good bases for obtaining highly accurate summaries. Moreover, our method replaces the usually bottom-up lexical chain construction with a top-down one, where first a single chain of disambiguated words is established and then it is divided in a sequence of many shorter lexical chains. The segmentation of text follows the sequence of lexical chains. Our methods of summarization control the length of the summaries by a process of scoring the segments. Thus, more material is extracted from the strongest segments.

The evaluation indicates acceptable performance when informativeness of summaries is considered. However, our methods have the potential to be improved: in CTT method we correspond a segment to a lexical chain. We intend to improve our scoring method of a segment by considering some recent method of scoring lexical chains (Ercan and Cicekli, 2008). Also, we intend to study how anaphora resolution could improve the lexical chains and the segmentation. We further intend to apply the presented methods to the corpus of texts DUC2002 and to evaluate them with the standard ROUGE method (for our experiments we didn't have the necessary human made summaries).

Acknowledgments

This work has been supported by PN2 Grant TD 400/2007.

References

Barzilay, R. and M. Elhadad (1999). Using lexical chains for Text summarization. In J. Mani and M. Maybury (Eds.), *Advances in Automated Text Summarization*. MIT Press.

Boguraev, B. and M. Neff (2000). Lexical Cohesion, Discourse Segmentation and Document Summarization. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*.

Corley, C. and R. Mihalcea (2005). Measuring the semantic similarity of texts. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor.

Ercan, G. and I. Cicekli (2008). Lexical cohesion based topic modeling for summarization. In *Proceedings of the Cicling 2008*, pp. 582–592.

Harabagiu, S. and D. Moldovan (1997). TextNet – a textbased intelligent system. *Natural Language Engineering 3*(2), 171–190.

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

Hovy, E. (2003). Text summarization. In R. Mitkov (Ed.), *The Oxford Handbook of Computational Linguistics*. Oxford University Press.

Mani, I. (2001). *Automatic summarization*. John Benjamins.

Marcu, D. (1997). From discourse structure to text summaries. In *Proceedings of the ACL/EACL '97 Workshop on Intelligent Scalable Text Summarization*, pp. 82–88.

Miller, G. (1995). WordNet: a lexical database for english. *Comm. of the ACM 38*(11), 39–41.

Morris, J. and G. Hirst (1991). Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics 17*(1), 21–48.

Orasan, C. (2006). *Comparative evaluation of modular automatic summarization systems using CAST*. Ph. D. thesis, University of Wolverhampton.

Radev, D., E. Hovy, and K. McKeown (2002). Introduction to the special issues on summarization. *Computational Linguistics 28*, 399–408.

Reynar, J. (1998). *Topic Segmentation: algorithms and applications*. Ph. D. thesis, Univ. of Penn.

Richie, D. (1991). The koan. In Z. Inklings (Ed.), *Some Stories, Fables, Parables and Sermons*, pp. 25–27.

Silber, H. and K. McCoy (2002). Efficiently computed lexical chains, as an intermediate representation for automatic text summarization. *Computational Linguistics 28*(4), 487–496.

Stokes, N. (2004). *Applications of Lexical Cohesion Analysis in the Topic Detection and Tracking Domain*. Ph. D. thesis, National University of Ireland, Dublin.

Tatar, D., A. M. G. Serban, and R. Mihalcea (2007). Text entailment as directional relation. In *Proceedings of CALP07 Workshop at RANLP2007*, Borovets, Bulgaria, pp. 53–58.

Tatar, D., A. Mihis, and D. Lupsa (2008). Text entailment for logical segmentation and summarization. In *13th International Conference on Applications of Natural Language to Information Systems*, pp. 233–244.

Tatar, D., G. Serban, A. Mihis, M. Lupea, D. Lupsa, and M. Frentiu (2007). A chain dictionary method for wsd and applications. In *Proceedings of the International Conference on Knowledge Engineering,Principles and Techniques*, pp. 41–49.

Tucker, R. (1999). *Automatic summarising and the CLASP system*. Ph. D. thesis, University of Cambridge.