Multilingual summarization system based on analyzing the discourse structure at MultiLing 2013

Daniel Alexandru Anechitei
"Al. I. Cuza" University of Iasi,
Faculty of Computer Science
16,General Berthelot St., 700483, Iasi,
Romania
daniel.anechitei@info.uaic.ro

Eugen Ignat
"Al. I. Cuza" University of Iasi,
Faculty of Computer Science
16,General Berthelot St., 700483, Iasi,
Romania
eugen.ignat@info.uaic.ro

Abstract

This paper describes the architecture of UAIC[1]'s Summarization system participating at MultiLing – 2013. The architecture includes language independent text processing modules, but also modules that are adapted for one language or another. In our experiments, the languages under consideration are Bulgarian, German, Greek, English, and Romanian. Our method exploits the cohesion and coherence properties of texts to build discourse structures. The output of the parsing process is used to extract general summaries.

1 Introduction

Automatic text summarization is a well studied research area and has been active for many years. In this paper, we describe the automatic text summarization system implemented by UAIC for participation at MultiLing 2013 single document track. Our approach to summarization follows the one presented in (Anechitei et al., 2013). The summarization architecture that this system uses includes two main parts that can be viewed in Figure 1. The text is passed to the language processing chain (LPC) which processes the data. As revealed from the figure each language has its own LPC. The LPC's, acts as a prerequisite for the summarization meta tool (SMT). In this paper we will focus more on the SMT engine, which is composed of four modules: anaphora resolution (AR), clause splitter (CS), discourse parser (DP) and the proper summarizer (SUM). The intermediate format between the modules consists of XML files. The summary of a text is obtained as a sequence of discourse clauses extracted from the original text, after obtaining the discourse structure of the text and exploiting the cohesion and coherence properties.

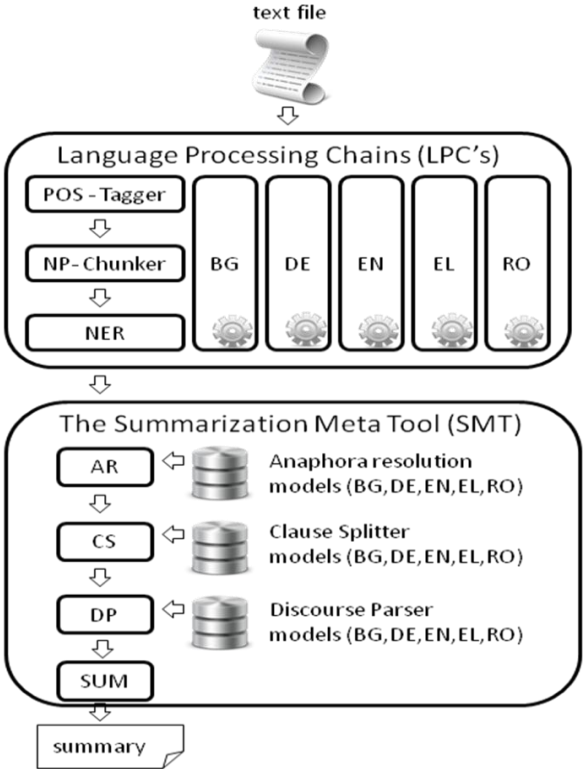

Figure 1: Summarization system architecture

2 Language Processing Chains

Every document is analyzed by the LPC in the following consecutive steps: sentence splitter, tokenizer, Part of Speech tagger, lemmatizer, Noun phrase extractor and Named entity recognizer. All tools are self-contained and designed

[1] University "Al. I. Cuza" of Iasi, Romania

Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization, pages 72–76,
Sofia, Bulgaria, August 9 2013.

to work in a chain, i.e. the output of the previous component is the input for the next component.

3 Anaphora Resolution

Anaphora resolution is one of the key steps of the discourse parser, by resolving anaphoric pronouns, automatically generated summaries may be more cohesive and, thus, more coherent. Calculating scores for references and transitions would be impossible without the proper identification of the co-referential chains.

Anaphora resolution is defined in (Orăsan et. al, 2008) as the process of resolving an anaphoric expression to the expression it refers to. The tool used for the anaphora resolution named RARE (Robust Anaphora Resolution Engine) uses the work done in (Cristea and Dima, 2001), where the process implies three layers (Figure 2):

- The text layer, containing referential expressions(RE) as they appear in the discourse;
- An intermediate layer (projection layer) that contains any specific information that can be extracted from the corresponding referential expressions.
- A semantic layer that contains descriptions of the discourse entities (DE). Here the information contributed by chains of referential expressions is accumulated.

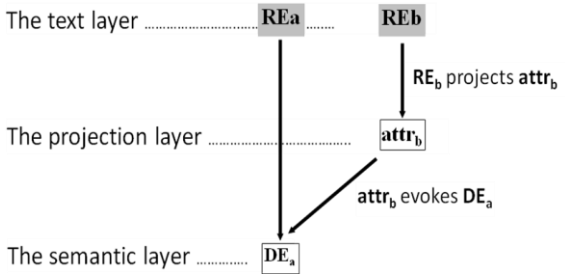

Figure 2: Three layers representation of co-referencing REs (Cristea and Dima, 2001)

The core of the system is language independent, but in order to localize it to one language or another it requires specific resources. These specific resources are as follows:

- **constraints** – containing the rules that match the conditions between anaphor and antecedent;
- **stopwords** – containing a list of stopwords;
- **tagset** – implies a mapping from the tagset used in the input file to a more simplified tagset used by the system.
- **window** – here is defined the length of the window where the antecedent should be looked for by the system.

The process of anaphora resolution runs as follows: The text is "read" from the left to right. When a new NP is found, a new RE is created and contains the morphological, syntactic and semantic features. All the features are tested using the constraints and it is decided whether the RE introduces a new discourse entity, not mentioned before, or it revokes one already mentioned.

4 Clause Splitter

Numerous techniques are used to recognize clause boundaries for different languages, where some are rule based (Leffa, 1988), and others are hybrid methods, like in (Parven et al., 2011) and (Orăsan, 2000), where the results of a machine learning algorithm, trained on an annotated corpus, are processed by a shallow rule-based module in order to improve the accuracy of the method. Our approach to discourse segmentation starts from the assumption that a clause is headed by a main verb, like "go" or a verbal compound, like "like to swim" (Ex.1). Verbs and verb compounds are considered pivots and clause boundaries are looked for in-between them.

Ex. 1 <*When I go to river*>< *I like to swim with my friends.*>

Verb compounds are sequences of more than one verb in which one is the main verb and the others are auxiliaries, infinitives, conjunctives that complement the main verb and the semantics of the main verb in context obliges to take the whole construction together. The CS module segments the input by applying a machine learning algorithm, to classify pairs of verbs as being or not compound verbs and, after that, applying rules and heuristics based on pattern matching or machine learning algorithms to identify the clause boundary. The exact place of a clause boundary between verbal phrases is best indicated by discourse markers. A discourse marker, like "because" (Ex.1), or, simply, marker, is a word or a group of words having the function to signal a clause boundary and/or to signal a rhetorical relation between two text spans.

Ex. 1 <*Markers are good*><*because they can give information on boundaries and discourse structure.*>

When markers are missing, boundaries are found by statistical methods, which are trained on explicit annotations given in manually built files. Based on the manually annotated files, a training module extracts two models (one for the CS module and one for the DP module). These models incorporate patterns of use of markers used to decide the segmentation boundaries and also to identify rhetorical relations between spans of text. The clauses act as terminal nodes in the process of discourse parsing which is described below.

5 Discourse Parser

Discourse parsing is the process of building a hierarchical model of a discourse from its basic elements (sentences or clauses), as one would build a parse of a sentence from its words (Bangalore and Stent, 2009). Rhetorical Structure Theory (Mann and Thompson, 1988) is one of the most popular discourse theories. In RST a text segment assumes one of two roles in a relationship: the nucleus (N) or satellite (S). Nuclei express what is more essential to the understanding of the narrative than the satellites. Our Discourse Parser uses a symbolic approach and produces discourse trees, which include nuclearity, but lacking rhetorical relation names: intermediate nodes in the discourse tree have no name and terminal nodes are elementary discourse units, mainly clauses. It adopts an incremental policy in developing the trees, on three levels (paragraphs, sentences and clauses) by consuming, recursively, one entire structure of an inferior level, by attaching the elementary discourse tree (*edt)* of the last structure to the already developed tree on the right frontier (Cristea and Webber, 1997). First, an *edt* of each sentence is produced using incremental parsing, by consuming each clause within the sentence. Secondly, the *edt* of the paragraph is produced by consuming each sentence within the paragraph. The same approach is used at discourse level by attaching the paragraph tree of each paragraph to the already developed tree. The criterion to guide the discourse parsing is represented by the principle of sequentiality (Marcu, 2000). The incremental discourse parsing approach borrows the two operations used in (L)TAG (*lexicalized tree-adjoining grammar*) (Joshi and Schabes, 1997): *adjunction* and *substitution*.

Adjunction operation (Figure 3) occurs only on the right frontier and it takes an initial or developing tree (D-tree$_{i-1}$), creating a new developing tree (D-tree$_i$) by combining D-tree$_{i-1}$ with an auxiliary tree (A-tree), by replacing the *foot node* with the cropped tree. This is done for each node on the right frontier resulting in multiple D-trees. Figure 3 depicts this idea.

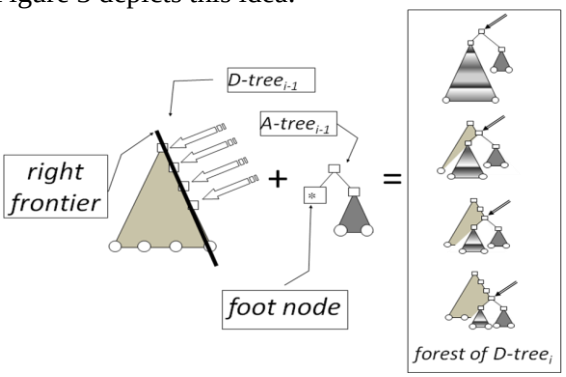

Figure 3: Adjunction operation

Substitution operation (Figure 4) replaces a placed node on a terminal frontier, called *substitution node*, with an auxiliary tree (Figure 14).

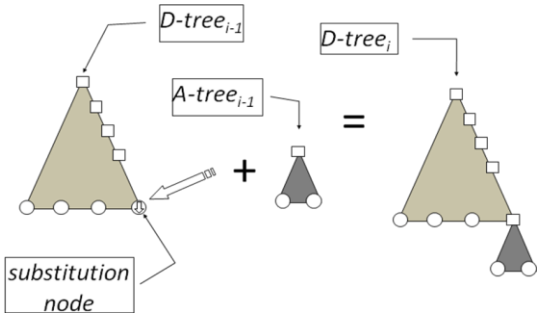

Figure 4: Substitution operation

The uses of different types of auxiliary trees (Figure 5) are determined by two factors:

- the type of operation in which are used: *alpha* and *beta* are used only for adjunction operations and *gamma* and *delta* for substitution operations;
- the auxiliary tree introduces or not an expectation: *beta* and *gamma* are auxiliary trees that raise an expectation and *alpha* an *delta* are auxiliary trees which do not raise an expectation.

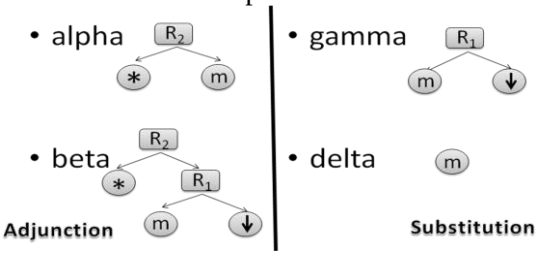

Figure 5: Types of auxiliary trees

At each parsing step there is a module which decides the type of the auxiliary tree between *alpha*, *gamma*, *beta*, *delta* (Anechitei et al., 2013) together with the relations type (R_1 and R_2, which can be N_N, N_S or S_N; the notation express the nuclearity of the child nodes: left one and the right one) by analyzing the structure which is processed (clause, sentence or paragraph). This module uses the compiled model described in previous section and doesn't produce a unique auxiliary tree for each structure but rather a set of trees.

At each level, the parser goes on with a forest of developing trees in parallel, ranking them by a global score (Figure 6) based on heuristics that are suggested by both Veins Theory (Cristea et al., 1998) and Centering Theory (Grosz et al., 1995). After normalizing the score for each heuristic, the global score is computed by summing the score of one heuristic with the corresponding weight. The weights were established after a calibration process.

$$GS^t = \sum_{i}^{N} s_i^h * w_i$$

Figure 6: Global score for each discourse tree

The trees used at the next step are only the best ranked trees. The aim of this filtering step is to reduce the exponential explosion of the obtained trees. For this task the threshold was set to *five best trees* from iteration to another and six (N=6) heuristics chosen in a way to maximize the coherence of the discourse structure and implicitly the coherence of the summary.

6 The Summarizer

The mentioned system produces excerpt type summaries, which are summaries that copy contiguous sequences of tokens from the original text.

The structure of a discourse as a complete tree gives more information than properly needed (at least for summarization purpose). By exploiting the discourse structure, we expect to add cohesion and coherence to our summaries. From the discourse structure we can extract three types of summaries: general summaries, entity focused summaries and clause focused summaries. For the summarization task we only extracted the general summary. The module that extracts the summaries (SUM) takes the tree of a discourse structure and produces a general summary, of a certain length, depending on the length of the computed vein (Cristea et al., 1998). As the task supposed summaries containing a maximum of 250 words and the summaries the system was providing were always bigger, a new scoring system was needed. This scoring system needed to shorten the summaries to under 250 words, yet keep as much coherence and cohesion as the system provided. For this end the scoring system took all the clauses from the vein and scored them as follows: in each clause the noun phrases were found, for each noun phrase a coreferential score was given. These scores are added and computed for each clause. The clauses were sorted and only the first N clauses were selected such as the maximum coherence was retained, where N is the number of the clauses so that the final summaries are below the word count threshold. The score for each noun phrase is given taking into account how big the coreference chain is.

7 Conclusion and Results

This year, the evaluation at MultiLing 2013 was performed automatically using N-gram graph methods, which were interchangeable in the single document setting. Below we provide the results based on average NPowER grades.

Lang	UAIC	Maryland (I)	Maryland (II)	Maryland (II)	Baseline
BG	1.538	1.600	1.593	1.600	1.310
DE	1.537	1.64	1.612	1.617	1.289
EL	1.560	1.501	1.513	1.494	1.314
EN	1.646	1.641	1.661	1.656	1.367
RO	1.627	1.655	1.679	1.680	1.346
	1.582	**1.607**	**1.611**	**1.609**	**1.325**

Table 1: Table with results

Table 1 shows the comparison between UAIC's system and Maryland's system, as it was the only other system, besides the baseline, that ran on the same 5 languages. Generally the results of both systems are close as the average figure shows. For our first participation the results are encouraging for this complex system, which has the possibility of running on multiple languages. Our future work should reside in the scorer of the summarizer, as the approach usually creates summaries bigger than 250 words.

References

Anechitei A. Daniel, Cristea Dan, Dimosthenis Ioannidis, Ignat Eugen, Karagiozov Diman, Koeva Svetla, Kopeć Mateusz and Vertan Cristina. 2013. *Summarizing Short Texts Through a Discourse-Centered Approach in a Multilingual Context*. In Neustein, A., Markowitz, J.A. (eds.), Where Humans Meet Machines: Innovative Solutions to Knotty Natural Language Problems. Springer Verlag, Heidelberg/New York.

Bangalore Srinivas and Stent J. Amanda. 2009. *Incremental parsing models for dialog task structure*, in Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics.

Cristea Dan and Webber Bonnie. 1997. *Expectations in incremental discourse processing.* In Proceedings of the 8th Conference of the European Chapter of the Association for Computational Linguistics.

Cristea Dan, Ide Nancy and Romary Laurent. 1998: Veins Theory: A Model of Global Discourse Cohesion and Coherence, in Proceedings of the 17th international conference on Computational linguistics.

Cristea Dan and Dima E. Gabriela. 2001. *An integrating framework for anaphora resolution*. In Information Science and Technology, Romanian Academy Publishing House, Bucharest, vol. 4, no. 3-4, p 273-291.

Grosz J. Barbara, Joshi K. Arvind and Weinstein Scott. 1995. *Centering: A Framework for Modelling the Local Coherence of Discourse*. Computational Linguistics, 21/2: 203-25.

Joshi K. Aravind and Schabes Yves. 1997: *Tree-Adjoining Grammars.* In G. Rozenberg and A.Salomaa, editors, Handbook of Formal languages.

Leffa J. Vilson. 1988. *Clause processing in complex sentences.* In Proceedings of the First International Conference on Language Resource and Evaluation, volume 1, pages 937 – 943, May 1998.

Mann C. William and Thompson A. Sandra. 1988. *Rhetorical structure theory: a theory of text organization*. Text 8(3):243–281.

Marcu Daniel. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. The MIT Press, November 2000.

Orasan Constantin. 2000. *A hybrid method for clause splitting in unrestricted English texts.* In Proceedings of ACIDCA, Corpora and Natural Language Processing, March 22-24, Monastir, Tunisia, pp. 129 – 134.

Orăsan Constantin, Cristea Dan, Mitkov Ruslan and Branco Antonio. 2008. *Anaphora Resolution Exercise – An Overview*. In Proceedings of LREC-2008, Marrakech, Morocco.

Parveen Daraksha, Sanyal Ratna and Ansari Afreen. 2011. *Clause Boundary Identification using Classifier and Clause Markers in Urdu Language*. Polibits Research Journal on Computer Science, 43, pp. 61-65.