Summarisation Through Discourse Structure

Dan Cristea[1,2], Oana Postolache[1,3], and Ionuţ Pistol[1]

[1] Al. I. Cuza University, Iaşi, Romania
{cristea, ipistol}@infoiasi.ro
[2] Institute for Theoretical Computer Science,
Natural Language Processing Group, Iaşi, Romania
[3] Computational Linguistics, Saarland University,
Saarbrücken, Germany
oana@coli.uni-sb.de

Abstract. In this paper we describe a method to obtain summaries focussed on specific characters of a free text. Summaries are extracted from discourse structures which differ from RST structures by the fact that the trees are binary and lack relation names. The discourse tree structures are obtained by combining constraints given by cue-phrases (resembling Marcu's method) with constraints coming from the exploitation of cohesion and coherence properties of the discourse (as proved by Veins Theory). The architecture of a summarisation system is presented on which evaluations intended to evidence the contribution of each module in the final result are performed and discussed.

1 Introduction

In this paper we describe an approach to discourse parsing and summarisation that exploits cohesion and coherence properties of texts. We built discourse structures that resemble the RST (Rhetorical Structure Theory [1]) trees, although ours are binary and lack relation names. Discourse tree building resembles the cue-phrase centred approach of Marcu [2] but adds to it constraints coming from the exploitation of the relation that is proved to exist by Veins Theory (VT) [3] between discourse structure and reference chains (a manifestation of cohesion), on the one hand, and between the global discourse structure and the smoothness of centering transitions (a manifestation of coherence) [4], on the other. The output of the parsing process is used to obtain excerpt-type summaries focussed on individual characters mentioned in the text. A combined, pipe-line/parallel/incremental, type of processing is employed.

The involved modules are POS-tagging, FDG-parsing, clause segmentation of sentences in clauses, construction of elementary discourse trees, detection of noun phrases (NPs), anaphora resolution (AR), discourse parsing and summarisation. To master the combinatorial explosion yield by different sources of ambiguity, a beam-search processing is employed. We present the architecture of a discourse parsing system and discuss the evaluation methodology. The final evaluation

A. Gelbukh (Ed.): CICLing 2005, LNCS 3406, pp. 632–644, 2005.

is realised by comparing the summaries output by the system against those contributed by human subjects.

Section 2 presents the overall method and the architecture of the system. Section 3 gives a quick overview on veins theory, which stays at the basis of the focussed summarisation method. Section 4 presents the method of incremental parsing and the module that assembles elementary discourse trees corresponding to sentences. Section 5 describes how the exponential explosion induced by different sources of ambiguity is controlled. In section 6 the corpus and the evaluation method are presented and section 7 discusses the results and synthesises some conclusions, limitations, and further work.

2 The Method

We call *focussed summary* on a character/entity X, a coherent excerpt presenting how X is involved in the story that constitutes the content of the text. Such summaries are of importance in information retrieval tasks from news or scientific papers when mentions of a certain entity are traced in a document. Note that a generic summary of a discourse sometimes will not include a desired character/entity if this entity appears only collaterally in the given discourse. Suppose, for instance, a drugs company interested to track in medical journals or scientific papers all mentions of a certain drug manufactured by them; neither extraction of the contexts of the drug mentions in the articles, nor generic summaries of the articles can be of help, as the intention is to know how is the drug mentioned *within the general topics of the articles*.

We describe the architecture of a system that combines a pipe-line style of processing the text with a parallel and an incremental one, with the aim to obtain an RST-like discourse structure that marks the topology and nuclearity, while ignoring the names of the rhetorical relations. Such trees are then used to compute focussed summaries on searched discourse entities. In the process of building discourse trees, we consider properties of the relationship between reference chains and the discourse structure as well as between global discourse structure and the smoothness of centering transitions. Both reference chains and centering transitions are related with veins expressions computed following the veins theory (VT) [3].

First, the text is POS-tagged, then a syntactic parser (FDG) is run over it. Further, the process is split into two flows: one that segments the sentences into *elementary discourse units* (*edu*s) and then constructs *elementary discourse trees* (*edt*s) of each sentence, and another that detects NPs and then runs an anaphora resolution engine to detect coreferential relations. Intermediate files in the processing flow are in the XML format. When two processes join, the resulted files are merged into a single representation. An *edt* is a discourse tree whose leaf-nodes are the *edu*s of one sentence. Sentence-internal cue-words/phrases trigger the constituency of syntactically *edt*s from each sentence [2], [5]. For each sentence in the original text a set of *edt*s is obtained. At this point a process that simulates the human power of incremental discourse processing is started. At any

moment in the developing process, say after n steps corresponding to the first n sentences, a forest of trees is kept, representing the most promising structures built by combining in all possible ways all *edt*s of all n sentences. Each such tree corresponds to one possible interpretation of the text processed so far. Then, at step *n+1* of the incremental discourse parsing, the following operations are undertaken: first, all *edt*s corresponding to the next sentence are integrated in all possible ways onto all the trees of the existing forest; then the resulted trees are scored according to four independent criteria, sorted and filtered so that only a fraction of them is retained (again the most promising after *n+1* steps). From the final wave of trees, obtained after the last step, the highly scored is selected. Summaries are then computed on this tree.

In [6] a general framework to resolve anaphors is proposed. We use this framework to integrate a model of coreference resolution that deals with most types of anaphors. Centering transitions scores are computed after AR is run, therefore after all references are solved. References and transitions, as well as heuristics for the proper development of a discourse tree, contribute with scores to the overall score of a developing discourse tree. These scores are then used to control the beam-search.

3 Veins Theory and Focussed Summarisation

Veins theory (VT) [3] is used in the described process to guide the incremental tree building and to synthesize summaries. VT makes two claims: emphasizes the close relationship between discourse structure and referentiality, as an expression of text cohesion, and generalizes Centering Theory (CT) [4] to the global discourse, as an expression of text coherence. Moreover, VT adds a view on summarization (consistent with [2]) and naturally reveals how focused summaries can be produced.

The fundamental intuition underlying an integrated account on discourse structure and accessibility in VT is that the RST-specific distinction between nuclei and satellites limits the range of referents to which anaphors can be resolved; in other words, the nucleus-satellite distinction, superimposed over a tree-like structure of discourse, induces a *domain of evocative accessibility* (*dea*) for each anaphor. More precisely, for each anaphor x in a discourse unit u, VT hypothesizes that x can be resolved by examining discourse entities from a subset of the discourse units that precede u. In this way VT reveals a "hidden" structure in the discourse tree, called *vein*. The notion of vein synthesizes observations on how references interact with the discourse structure represented as an RST tree in which names of relations were ignored (we will call such a simplified representation an RST-like tree). Considering the hierarchical organization given by the tree structure and the principle of compositionality [2], which induces recursively long-distance relations between *edu*s, these observations can be stated as follows:

- a right satellite or a nucleus can refer its left nuclear sibling;
- a right nucleus can refer its left satellite;

- in a combination $n_1\ s_1\ s_2$, with s_1 and s_2 satellites of the nucleus n_1, s_1 is not accessible from s_2;
- in a combination $n_1\ s_1\ n_2$, with s_1 a satellite of the nucleus n_1 and n_2 a right nuclear sibling of n_1, s_1 is not accessible from n_2;
- a nucleus blocks the reference from a right satellite to a left satellite, therefore in a combination $s_1\ n_1\ s_2$, with s_1 and s_2 satellites of the nucleus n_1, s_1 is not accessible from s_2.

The vein expression of an *edu* u is a list of *edu*s of the discourse, including u, which is meant to express the sequence of units that are significant to understand u **in the context of the whole discourse**.

VT classifies references into three categories, in accordance with the way they align along the veins. An anaphor, belonging to an *edu* u_2, is said to issue a **direct reference**, if its linearly most recent antecedent belongs to an *edu* u_1 that is included in u_2's vein. Under the same notations, it issues an **indirect reference** if u_1 does not belong to u_2's vein, but there is a more distant antecedent, say belonging to an *edu* u_0, and u_0 is placed on u_2's vein. If the backward-looking reference chain of the anaphor does not intersect the vein of the anaphor's *edu*, we have an **inferential reference**. VT conjunctures on two types of anaphoric processes: **evocative** (or **immediate**)and **post-evocative** (or **inferential**). The evocative processes are most frequent, are rapid and can be realised by any referential means, including those as fragile as empty pronouns. They make the discourse fluid and increase the text cohesion. An evocative anaphora occurs anytime the backward-looking chain of referential links having the right-most end in the current anaphor intersects at least once the vein expression of the *edu* the anaphor belongs to (the cases of direct and indirect references). This means that the antecedent can be recuperated looking to the left only in the sub-discourse obtained by concatenating the *edu*s in the vein expression of the current anaphors *edu*. The post-evocative anaphorae are less frequent, induce more inferential load on the reader (hearer) and make use of strong referential means (like proper nouns, for instance). A post-evocative anaphora is one in which there is no *edu* of the anaphor's referential chain which belongs also to the anaphor's vein expression (the case of the inferential reference).

A corollary of VTs claims is that the text obtained by the concatenation of the spans indicated in the vein expression of an *edu* is a sub-discourse that gives a summary of the whole discourse, focused on that particular unit. Now, suppose one discourse entity is traced and a summary focused on that entity is desired. If there is only one *edu* in which the entity is mentioned, the vein expression of that *edu* gives a very well-focused summary of the entity. A problem appears if the entity is mentioned in more than just one *edu*. Because there is no a-priory reason to prefer one of the focused summaries obtained in this way to any of the others, it is clear that a combination of the vein expressions of each *edu* in which the entity is mentioned should be considered. We have proposed more methods [5] of building a final summary from the collection of particular summaries. The first method takes the vein expression of the lowest node of the tree that covers all units in which the entity is mentioned. Since the

length of a vein expression is proportional to the deepness of the node in the tree structure, this method results in shorter summaries. The second method considers that particular summary (vein expression) which sums most of the mentions of the entity. The third method simply takes the union of all vein expressions of the units that mention the entity in focus. Finally, the fourth method builds a histogram from all vein expressions of the units mentioning the focussed entity and selects all units above a certain threshold. The last two methods are not in themselves vein expressions, and therefore are more prone to incoherent summaries than the first two methods, the last one being the most exposed. In our experiments till now we have used only the first method.

4 Incremental Parsing

The basic step in an incremental discourse parser is the integration of an elementary discourse tree (*edt*), which corresponds to a sententence, int the tree representing the discourse structure of the discourse parsed so far. By doing this we will obtain discourse trees in which to each sentence corresponds one node of the discourse structure covering exactly the sentence's span ([7] have shown that in 95% of the cases this is true). The operations applied at each step during the incremental processing is *adjunction* on the right frontier [8]. Cue-words and cue-phrases (markers) are connectives having a signalling function on: the nuclearity of the *edu*s they interconnect, the form of the *edt* they belong to, and the place on the right frontier of the developing tree where an *edt* is to be adjoined. Subordinate connectives, like *just*, *as*, *although*, *as long as*, *whenever*, *because*, etc., link subordinate clauses (satellite structures) onto regent clauses (nuclear structures), while coordinate connectives, like *and*, *or*, etc., usually link sibling nuclear structures. There are also frequent cases when connectives miss completely. Different patterns of arguments for markers have been manually selected from a corpus. Fig. 1 depicts some cases (the dots suggest the nuclearities of their arguments). There are frequent cases when the same marker has more than one argument pattern.

As constraints to build syntactically correct trees we have used the rules described in [5]. Such constraints configure *edt*s in which inner nodes are labelled with markers and leaf-nodes with *edu* labels. Each node of the tree is also marked by a nuclearity function with *n* (for nuclear) or *s* (for satellite) so that at each level, between the two descendents of an inner node, at least one is marked *n*, and the root of an *edt* is always marked *s*. Since the number of inner nodes of a binary tree with *t* leaf-nodes is *t-1*, for an *edt* to be completely determined it needs a

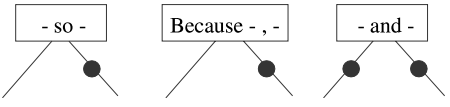

Fig. 1. Argument patterns of cue-phrases

number of cue-words, as inner *edt* nodes, with one less than the number of *edu*s. For such reasons we apply heuristics to add dummy markers where missing. Dummy markers are empty strings similar to *and* with both arguments labelled as nuclear (the implicit assumption is that a satellite is always announced by a realised marker). The incremental parsing in [5] is deterministic. Heuristics help, at each step, to adjoin the current *edt* in that place of the right frontier of the developing tree which maximizes the chances to arrive at a correct final analysis. Instead, our analysis does not go deterministically. At each step, all possible trees resulted from the application of marker argument-structure patterns and syntactic constraints are generated and then are adjoined in all possible positions of the right frontier of the developing tree. To control the exponential explosion induced by this luxurious behaviour we implemented a beam-search-like process.

5 The Beam-Search Control

Any beam-search-like process depends heavily on a scoring function able to appreciate the relevance of the objects produced at intermediate steps, and which are successively detailed or improved until a final object, supposed to satisfy the goal, is obtained. In this section we explain our scoring criteria. In [9] an empirical evaluation of VT's conjectures is described. Experiments drawn on corpora annotated to both discourse structure (RST) and coreference have shown that VT's conjectures are generally correct. The authors of VT report that 87.1% of all references they found in the investigated corpus are direct, and 8.5% are indirect. The rest of 4.4% escape the predictions of VT, some being classified as of a pragmatic type (not needing an antecedent in order to be understood) [3]. However, an important aspect is that exceptions align their frequencies per types with their evoking power, as folows: pragmatic – 56.3%, proper nouns – 22.7%, common nouns – 16.0%, pronouns – 5.0%. Following [10], the *evoking power* of each of these types of REs decreases as we move to the right in the list. Pragmatic references are those which refer to entities that can be assumed as part of general knowledge, such as *the Senate* or *our* in the phrase *our streets*. The descending order of the types of the expressions disobeying VT suggests that pragmatic references are easily understood without an antecedent while proper nouns and common noun phrases are understood less and less. At the other extreme, pronouns have very poor evoking power: a message emitter employs them only when s/he is certain that the structure of the discourse allows for an easy recuperation of the antecedent in the message receiver's memory. Except for the cases where a pronoun can be understood without an antecedent (as in the example with *our* in *our streets*), the use of a pronoun referring an antecedent that is outside the *dea* should produce an invalid message. Since the detection of pragmatic references requires knowledge that goes beyond the possibilities of our sources, we considered only proper nouns, comon nouns and pronouns for the scoring criterion based on references.

To score *references in relation with veins* we have given the values 2, 1 and 0 for the values **direct**, **indirect** and **outside vein**, respectively. Then, to score

the *anaphor type* we have given the values 3, 2 and 1 for the following categories of anaphors: **pronoun**, **common noun** and **proper noun**, respectively. Then we have multiplied these scores for each anaphor, allowing each anaphor to contribute to the general score of the tree with a value between 0 and 6, with 0 meaning that any of its antecedents are outside the *dea* of the unit of the anaphor, and 6 in case of a pronoun whose most recent antecedent is on the *dea* of the unit the anaphor belongs to. This is the s_r section of the score (see below).

The second tree-scoring criterion used the coherence conjecture of VT. Following [3], we let each unit to contribute with a score between 0 and 4, depending on the type of centering transition between the current unit and the previous unit in the vein expression, in ascending order of smoothness: **no C**$_b$, **abrupt shift**, **smooth shift**, **retaining** and **continuing** [4]. As will be shown below, the score formula is designed to keep track of the relationship between references and structure. This is the section s_c of the score (see below). The overall contribution in the score of a tree coming from VT represents the s_1 section of the score formula, and has the following form:

$$s_1 = \sum_{u \in D} \left(w_1 \sum_{x \in RE_u} \frac{s_r^x}{6} + w_2 \frac{s_c^u}{4} \right) \tag{1}$$

where u is an *edu*, D represents the whole discourse, x is an anaphor, RE_u is the set of the anaphors belonging to unit u which have antecedents outside that unit, s_r^x is the referential score contributed by the anaphor x and s_c^u is the centering score contributed by the unit u. The two weights w_1 and w_2 sum-up to 1 and are iteratively computed to accommodate optimally the score scheme to the expected results.

During the experiments we have noticed a tendency of the parsing trees to be skewed downward and to the right (a tree with this particular shape corresponds to a discourse in which each *edu* addsa detail to the preceding one, while a tree completely skewed upward and to the right corresponds roughly to a discourse in which each *edu* adds a detail to the initial *edu*. To balance this tendency we scored better an adjunction of an *edt* on the upper part of the right frontier of the developing tree than on the lower part. The contribution of this criterion represents the s_2 section in the score formula (see below).

Section s_3 of the score formula is thought to penalize too many nuclear nodes in the final tree. A tree that has only nuclear nodes is a flat structure, but between the two daughters of a node at least one should be nuclear. So, s_3 is the fraction between the number of satellites and the total number of nodes of the tree.

Finally, the last section of the score, s_4, reflects the quality of the *edt*s which are build from sentences. Each *edt* is compared against the structure returned by the FDG parser (only for English) with respect to the nuclearity of the *edu*s (0.5) and the identity of the sibling node in the structure (0.5) and then we average the sum on the number of *edu*s in the segment.

In principle, at each step of the search we have a fixed number N of developing trees and to each of them we adjoin in all possible ways all computed *edt*s. The

score of each new developing tree obtained as such is calculated as the product s_1 * s_2 * s_3 * s_4. Then we sort all these trees in the descending order of their scores and we retain for the next step again the first N best rated trees. At the end of the run, the best scored final tree gives the discourse structure.

6 Corpus and Evaluation

We have done parallel experiments on both Romanian and English. As a test we have used a fragment summing up 812 words from G. Orwell's novel "1984" in the English version and 863 words in its Romanian equivalent .

We believe that the evaluation of a complex NLP system should follow a procedure that facilitates an easy inventory of the depreciation of performance along the processing chain. This way, the identification of critical points of the system is straightforward and repairing can be focussed towards the points of maximum trouble. In this section we show how we use such a technology in order to evaluate our summarizer for both English and Romanian. The overall processing flow of the system and the points where the "temperature" is measured are depicted in Fig. 2. Early processing phases, as POS-tagging and FDG-parsing are considered included in the input in this scheme. Processing modules are indicated in light grey rectangles, evaluation results in dark squares, and files in rounded rectangles: those which are pure outputs of processing modules - in white, and those influenced in any way by a gold-standard - shadowed. The names of the files indicate their origin, so, for instance `np-seg-gold-ar-edt-tree-test` is a file that records a gold-standard (gold) of manually annotated noun-phrases (np) and *edu*s (seg), as well as the results (test) of runing the AR-module (ar), the edt-detector module (edt) and the discourse parser module (tree). Also, `sum-gold` and `all-test` are the two most distant final files, recording respectively the gold-standard of summary and the output of a complete and pure (no human intervention) processing chain.

All initial gold standards, `seg-gold`, `np-gold` and `np-ar-gold` have been created by master students in Computational Linguistics, while the `sum-gold` file was build with the help of a class of 91 terminal year undergraduate students in Computer Science, during an NLP examination. They received the initial text in which *edu*s were already marked and numbered and were asked to indicate 4 summaries by writing down sequences of discourse unit numbers: a general summary of the whole text of about 20% reduction rate and three summaries focussed on different characters mentioned in the text (*Winston's mother*, *Winston's sister* and *the girl with black hair*). For each *edu* of the original text we counted the number of times this *edu* was included in any students' summaries. As such, a histogram resulted, with the sequence of *edu* numbers on the x-axis and the frequency of mentioning on the y-axis. Then we considered a sliding horizontal threshold on this histogram, and accepted as belonging to the golden summary all units whose corresponding frequencies were above the threshold. During tests we have established the threshold to a number of hits of 20, which resulted in a gold-summary of length 30 *edu*s.

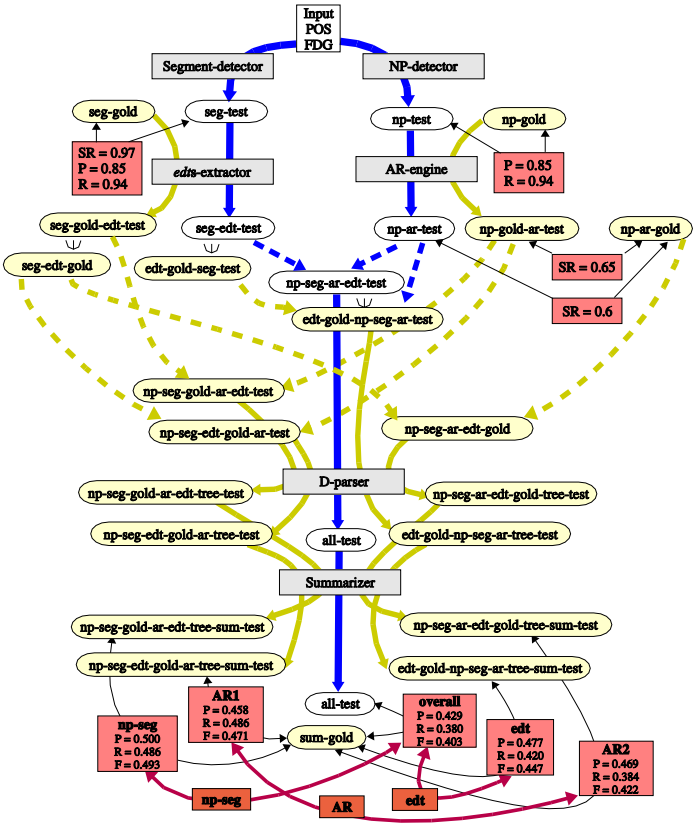

Fig. 2. Processing and evaluation points

Fig. 2 shows the processing flow and results for the implementation running English texts. In the upper part of the diagram the evaluation points are meant to determine the behaviour of the segment-detector, the NP-detector and the AR-engine, independent of the overall summarization task of the system. In the figure, P stands for precision, R for recall and SR for success rate, conforming to [11]. Precision and recall in the case of segment-detector have been computed in terms of segment borders, while success rate as the number of words correctly assigned to segments (belonging to *edu*s around the same main verb), divided by the total number of words.

As Fig. 2 shows we do not have a gold standard for discourse structure (a file `tree-gold` is absent). To evaluate our trees we used instead summaries, easier to acquire than RST-like annotations of discourse structure. If summaries extracted automatically, as by-products of a discourse parsing process, resemble those indicated by human subjects, then we should a high degree of confidence that the structures themselves reflect with enough accuracy the text content.

As baseline for our general summaries evaluation we have used the summary produced by MS Word on the same text. As baseline for the three focussed summaries we selected all sentences containing the expressions *his mother*, *his sister* and *girl*. Example 1 displays part of the text under experiment, on which the gold general summary is in boldface and the automated summary in italics.

Example 1. ***Winston was dreaming of his mother.***

He must, he thought, **have been ten or eleven years old when his mother had disappeared.** *She was a tall, statuesque, rather silent woman with slow movements and magnificent fair hair.* **His father he remembered more vaguely as dark and thin, dressed always in neat dark clothes** (Winston remembered especially the very thin soles of his father's shoes) and wearing spectacles. **The two of them must evidently have been swallowed up in one of the first great purges of the fifties.**

At this moment his mother was sitting in some place deep down beneath him, with his young sister in her arms. ***He did not remember his sister at all, except as a tiny, feeble baby, always silent, with large, watchful eyes.*** **Both of them were looking up at him. They were down in some subterranean place – the bottom of a well, for instance, or a very deep grave** – but it was a place which, already far below him, was itself moving downwards. ***They were in the saloon of a sinking ship, looking up at him through the darkening water.*** There was still air in the saloon, they could still see him and he them, but all the while they were sinking down, down into the green waters which in another moment must hide them from sight for ever. ***He was out in the light and air while they were being sucked down to death****, and they were down there* because he was up here. He knew it and they knew it, and he could see the knowledge in their faces. *There was no reproach either in their faces or in their hearts, only the knowledge* that they must die in order that he might remain alive, and that this was part of the unavoidable order of things.

He could not remember what had happened, but he knew in his dream that in some way the lives of his mother and his sister had been sacrificed to his own. *It was one of those dreams which*, while retaining the characteristic dream scenery, are a continuation of one's intellectual life, *and in which one becomes aware of facts and ideas* which still seem new and valuable after one is awake. **The thing** that now suddenly struck Winston **was that his mother's death, nearly thirty years ago, had been tragic and sorrowful in a way that was no longer possible**. *Tragedy, he perceived, belonged to the ancient time, to a time* when there was still privacy, love, and friendship, *and when the members of a family stood by one another without needing to know the reason.* His mother's memory tore at his heart because she had died loving him, when he was too young and selfish to love her in return, and because somehow, he did not remember how, she had sacrificed herself to a conception of loyalty that was private and unalterable. ***Such things****,he saw,* ***could not happen today.***

7 Discussions and Conclusion

As seen in Fig. 2 the segment-detector behaves satisfactory. A less good precision but very good recall was obtained also for the NP detector. A significant deterioration of the results are expected to occur following the AR-phase since the extreme extravagance of a free text as Orwell's novel and the need to trace

Table 1. Statistics of the edt-extractor

No of *edus*	No of sentences of this length	No of generated *edt*s per sentence
1-3	25	1-4
4-5	9	5-28
6	1	42

at once all types of anaphors made resolution of the coreferring anaphora a very difficult task. Comparing the two SR values (0.65 versus 0.6) one can perceive the influence of the NP-detector on the deterioration of the performance of the AR-engine. This behaviour is conformant to the expectations since NPs are the referential expressions that are worked out by the AR-engine. The *edt*s-extractor computed *edt*s as shown in Table 1.

We tested our discourse parser (D-parser in Fig. 2) over the set of 83 *edu*s which were grouped in 35 sentences in both `seg-gold` and `seg-test`.

To master the tree explosion we have used a slightly different threshold policy than the one described in section 5: after each step of the D-parser we have kept only the most promising trees whose combined scores range in a threshold of zero under the best score (tie-vote on the maximum). Using this policy, the maximum number of trees generated in any of the 35 steps was 320.

To learn the optimum weight values of parameters w_1 and w_2 of formula (1) we have run 10 times the whole parser modifying at each step w_1 by 0.1 (remember that w_2 = 1 - w_1). The final results of the general summaries are shown in Fig. 2. For comparison, the MS Word-baseline for the general summary was rated with a precision of 0.222, a recall of 0.176 and an F-measure of 0.197. Also, the best student general summary was rated with a precision of 1.00, a recall of 0.679 and an F-measure of 0.801. The implementation was done in Java. The interested reader can consult documentation and perform experiments with modules described in this paper at the following adresses: AR-engine at `www.coli.uni-sb.de/~oana/rare` and Discourse Parser and Summarizer at `www3.infoiasi.ro/~ipistol/parser`.

Different black boxes displaying recall (R) and precision (P) and F-measure (F) values in the lower part of Fig. 2 show different evaluations made over the summarisation system by comparing outputs in which part of the work is done manually and part of it automatically against the summary gold standard file `sum-gold`:

- `overall` – evaluates the all-automatically obtained output file `all-test`;
- `edt` – evaluates the output corresponding to an input in which elementary trees of sentences have been contributed manually;
- `np-seg` – evaluates the output corresponding to a manual detection of NPs and segmentation;

- `AR1` – evaluates the output corresponding to an input in which all the following steps have been performed manually: detection of NPs, segmentation, and elementary tree detection.
- `AR2` – supplementary to `AR1`, has also the anaphora resolution process manually annotated.

As seen, the results are above the baseline, although the values are still low. The evaluation operated at different point in the processing chain validate the expectations: the more gold components we incorporate, the more accurate are the results. We could also estimate the impact of the component modules on the summaries by counting the differences between R and P values at the edges of the thick arrows: NP-detector + segment-detector, as the difference between `np-seg` and `overall` values = 0.090; *edt*s-detector, as the difference between `edt` and `overall` values = 0.044, and AR-engine, as the difference between `AR2` and `AR1` values = 0,049. So, it seems that low level processes, as detection of NPs and segmentation influence more the summarization results than high level processes as *edt*-detection and AR resolution. The results on Romanian are still under development, but we expect to be under the ones for English because of the lack of an FDG parser.

The following aspects will make the subject of further work: retraining of the AR and segmentation processes with different heuristics, implementation of the substitution operation in incremental discourse parsing, and the improvement of the performances of the individual modules, and implementation of different focused summarisation criteria by exploiting the vein expression, as described at the end of section 3.

Acknowledgements

Our thanks go to our students who have done the manual annotations and have produced the summaries that helped to draw the final evaluation. Special thanks go to our colleagues from the Laboratory of Computational Linguistics of the University of Wolverhampton who have kindly provided the FDG annotated version of the "1984". Part of the work reported in this paper was performed while the first author was in a visiting research stage at ITC-IRST Trento.

References

1. Mann, W.C., Thompson, S.A.: Rhetorical structure theory: A theory of text organization. Text **8:3** (1988) 243–281
2. Marcu, D.: The Theory and Practice of Discourse Parsing and Summarization. The MIT Press (2000)
3. Cristea, D., Ide, N., Romary, L.: Veins theory: A model of global discourse cohesion and coherence. In: Proceedings of COLING/ACL, Montreal/Canada (1998)
4. Grosz, B.J., Joshi, A.K., Weinstein, S.: Centering: A framework for modelling the local coherence of discourse. Computational Linguistics (1995)

5. Cristea, D., Postolache, O., Puşcaşu, G., Ghetu, L.: Local and global information exploited in producing summaries. In: Proceedings of the International Symposium on Reference Resolution and Its Applications to Question Answering and Summarisation, Venice/Italy (2003)
6. Cristea, D., Dima, G.E.: An integrating framework for anaphora resolution. Information Science and Technology **4** (2001) 273–291
7. Şoricuţ, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: Proceedings of HLT-NAACL, Edmonton, Canada (2003)
8. Polanyi, L.: A formal model of the structure of discourse. Journal of Pragmatics **12** (1988) 601–638
9. Cristea, D., Ide, N., Marcu, D., Tablan, V.: An empirical investigation of the relation between discourse structure and co-reference. In: Proceedings of COLING, Saarbücken/Germany (2000) 208–214
10. Gundel, J., Herberg, N., Zacharski, R.: Cognitive status and the form of referring expressions in discourse. Language **69** (1993) 274–307
11. Mitkov, R.: Anaphora Resolution. Longman, London (2002)