How Could Veins Speed Up The Process Of Discourse Parsing

Elena Mitocariu[1], Daniel-Alexandru Anechitei[1], Dan Cristea[1,2]
[1] "Alexandru Ioan Cuza" University of Iasi, Faculty of Computer Science
16, General Berthelot St., 700483 – Iasi, Romania
[2] Romanian Academy, Institute for Computer Science
{elena.mitocariu, daniel.anechitei, dcristea}@info.uaic.ro

Abstract

In this paper we propose a method of reducing the search space of a discourse parsing process, while keeping unaffected its capacity to generate cohesive and coherent tree structures. The parsing method uses Veins Theory (VT), by developing incrementally a forest of parallel discourse trees, evaluating them on cohesion and coherence criteria and keeping only the most promising structures to go on with at each step. The incremental development is constrained by two general principles, well known in discourse parsing: sequentiality of the terminal nodes and attachment restricted to the right frontier. A set of formulas rooted on VT helps to guess the most promising nodes of the right frontier where an attachment can be made, thus avoiding an exhaustive generation of the whole search space and in the same time maximizing the coherence of the discourse structures. We report good results of applying this approach, bringing in a significant improvement in the discourse parsing process.

Keywords: veins theory, discourse structure, incremental discourse parsing, reduction of the execution time

1. Introduction

Discourse parsing has traditionally dealt with short texts such as newspapers and articles, but new approaches such as analysis of lexical repetition (Boguraev and Neff, 2000), identification of topics (Utiyama and Isahara 2001) or using of thematic hierarchy of text (Nakao, 2000), takes into consideration also longer texts.

Discourse parsing systems combine lexical, syntactic and semantic features to generate representative discourse trees. But discourse parsing is often intended to work on large texts, characterized by complex analyses, which are normally obtained over expensive processing time. Different theories present various types of discourse structure representations such as trees (Mann and Thompson, 1988) and graphs (Asher and Lascarides, 2003).

Discourse theories divide the text in spans which are connected through different type of relations. Rhetorical Structure Theory (RST) and Segmented Discourse Representation Theory (SDRT) (Asher, 1993) postulate possible asymmetries of the relation arguments: when two text spans are in a certain relation one with respect to the other, one can play a "subordinate" (less important) role relative to the other. This asymmetry is expressed in RST as a distinction between nuclei and satellites and in SDRT as a distinction between coordinating and subordinating relation links (Danlos, 2008).

Another model of discourse structure, Veins Theory (VT) (Cristea et al., 1998a), places centrally the nuclearity of relations, revealing hidden structures on discourse trees, called veins, which emphasise the manifestation of cohesion and coherence properties of discourses. According to VT, a discourse tree correctly characterizes a text if two criteria are maximally realized on the set of veins corresponding to the discourse units:

- a cohesion criterion – computes a score associated to the resolution of anaphors on antecedents placed on veins;
- a coherence criterion – computes a global smoothness score of a discourse by summing up Centering transitions scores (Grosz et al., 1995) in which the utterances (discourse units) are ordered hierarchically, i.e. along veins, not linearly.

The incremental parsing technique, described in (Anechitei et al., 2013) and rooted on an approach introduced in (Cristea and Webber, 1997) and (Cristea et al., 1998a), uses these two criteria to guide a beam search process in a space of partially developed discourse trees. At each step, the parser retains the most promising *N* trees among those obtained after adjoining an auxiliary tree on the right frontier of the developing structure, where *N* is determined by the space-speed limitations of the machine accommodating the parser.

During the development of the discourse tree, two principles are consistently observed at each step of the incremental process:

a) the Sequentiality Principle (Marcu, 2000);
b) the Right Frontier Constraint, stated empirically by many scholars, as Webber (1991), (Afantenos and Asher, 2010).

Our approach applies to binary trees and takes into account the right frontier constraints (RFC) (Webber, 1991; Cristea, 2005). As was demonstrated in (Afantenos and Asher, 2010) RFC can be formulated for SDRT, which makes our model extensible on discourse parsing systems that perform on SDRT representations as well. In this paper we show how a set of formulas rooted on Centering Theory (CT) (Grosz et al., 1995) and VT aid maximizing the coherence of a discourse tree. We present the results of applying our method on a discourse parser system that generates binary trees in which leafs are elementary discourse units (*edus*), such as clauses or short sentences, and internal nodes represent larger text spans. RST uses a labeling function that attaches relation names and nuclearities to its inner nodes, while VT ignores names of relations. Because of this simplified representation, one can say that VT is included in RST (Mitocariu et al., 2013). In the next sections we briefly

present a discourse parser and both theories (CT and VT). Then we will focus on describing the set of formulas and how they can be applied to plan coherent discourse structures while also reducing the search space of attaching nodes on the RF. Finally we analyze the results and draw some conclusions.

2. Discourse Parsing

Discourse structures have a central role in several computational tasks, such as summarization, question-answering (QA), information extraction (IR), etc. Discourse parser systems are developed taken into account different features. Some are based on semantic proprieties and other use syntactic characteristics of texts. Usually, discourse parsing systems combine these features to generate representative discourse trees, which, among others, can root approaches aiming at summarizing texts. We believe that a good summary extracted from a discourse structure is one that, besides the fact that it must give a shorter overview over the text, should preserve the qualities of being cohesive and coherent. This is why the use of referential expressions is of primary interest for obtaining coherent discourse structures.

2.1 Centering

Centering Theory (CT) is one of the most influencing theories in explaining coherence properties of discourses. It estimates coherence between two adjacent utterances by placing transitions on a scale of 5 layers, from the most easiest to interpret (CONTINUATION) to the most difficult (NO CB). The classification of transitions into five types resides on the notion of center (as semantic representations of referential expressions) and their sharing between adjacent utterances. Each utterance (discourse unit) sets a list of forward-looking centers – $Cf(U_n)$ – as the centers realized in the current utterance. For each discourse unit other than the initial one, a backward-looking center $Cb(U_n)$ can be determined, as the first center of $Cf(U_{n-1})$ which exists also in $Cf(U_n)$. This definition allows also a lack of Cb, when the two consecutive utterances do not share a common center. From the elements of the Cf list the highest-ranked member is called the preferred center $Cp(U_n)$. The five different types of transitions between pairs of successive discourse units (U_n, U_{n+1}) are the following:

- **CONTINUING** (score 4):

$Cb(Un+1) = Cb(Un)$ OR $Cb(Un) = NULL$
$Cb(Un+1) = Cp(Un+1)$

- **RETAINING** (score 3):

$Cb(Un+1) = Cb(Un)$ OR $Cb(Un) = NULL$
$Cb(Un+1) \neq Cp(Un+1)$

- **SMOOTH SHIFTING** (score 2):

$Cb(Un+1) \neq Cb(Un)$
$Cb(Un+1) = Cp(Un+1)$

- **ABRUPT SHIFTING** (score 1):

$Cb(Un+1) \neq Cb(Un)$
$Cb(Un+1) \neq Cp(Un+1)$

- **NO Cb** (score 0).

By averaging the transitions over the whole discourse, a global Centering score is obtained (Figure 1) (Cristea et al., 1998a) which reflects the coherence of the text. Here, by TScore we denote the transition score between each two consecutive utterances.

$$Centering\ Score = \frac{\sum TScore}{N}$$

Figure 1: General Centering Score

2.2 Veins Theory

Veins Theory makes two important claims: the first regards discourse cohesion, the second – discourse coherence. VT extends CT from a local to a global level. It takes from RST the binary tree representations of discourse structures and the notions of nucleus and satellite, but leaves out the names of relations.

The leaves of the discourse tree represent elementary discourse units and the internal nodes, including the root, represent larger spans of text. When two nodes have a common parent it means that they are in an anonymous discourse relation and in this relation they have a nuclear role (N) or a satellite role (S). A nucleus is more important than a satellite, such that if a nucleus would be eliminated, the text would loose coherence, but if a satellite would be eliminated, it would loose some details but its coherence would remain unaffected. The material nodes (elementary discourse units, leafs) are supposed to be identified by individual labels. VT introduces two expressions (which represent sequences of material nodes), called *head* and *vein*, computed as follows:

The *Head* expression of a node is meant to identify the sequence of the most salient material nodes in the span covered by that node. Head expressions are computed bottom-up:

– if the node is a leaf, its head expression is its label;

– else, the head expression is the concatenation of the head expressions of its nuclear children.

The *vein* expression of a node *n* is meant to signify the sequence of elementary discourse units which are sufficient to understand the span covered by the node *n* in the context of the whole discourse. In the definition of vein expressions the following functions, taking as arguments sequences of labels, are used:

– *seq*, returns the right frontier reordering in the left to right order of the sequence given by of the concatenation of its arguments;

– *mark* returns the same symbols as in its argument, but marked in some way (for example, between parentheses or primed);

– *simpl* eliminates all marked symbols from its argument.

With these, vein expressions of all nodes in the discourse tree are computed top-down, as follows (Cristea et al., 1998a):

– the *vein* expression of the root node is its head expression;

– if the node is a nucleus and its parent's vein expression is *v*, then:

  • if the node has a left satellite sibling with head *h*,

then its vein expression is *seq (mark(h), v)*;

• else, *v*;

– if the node is a satellite with the head *h* and its parent's vein expression is *v*, then:

• if it is a left daughter, then its vein expression is *seq(h,v)*;

• else its vein expression is *seq(h,simpl(v))*.

2.3 Methodology

The method we describe in this paper was applied on a discourse parsing system that runs on multiple languages (Bulgarian, German, Greek, English, Romanian and Polish) and produces summaries (Anechitei et al., 2013). The system architecture process the text in the following consecutive steps: sentence splitting, tokenization, part-of-speech tagging, lemmatization, noun phrase extraction, named entity recognition, anaphora resolution, clause splitting and discourse parsing. Since it summarizes thousands of documents per day, there is a demanding necessity to improve its efficiency. The discourse parser applies an incremental strategy in developing the trees, at each step observing the principle of sequentiality (Marcu, 2000) and the RFC (Webber, 1991).

The incremental development of a discourse tree is performed by continuously applying two operations inspired by Tree Adjoining Grammars (Joshi and Schabes, 1997): *adjunction* and *substitution*. According to Cristea and Webber (1997), out of the two, only adjunction allows for more options at each step (the whole generalized right frontier positions), while the substitution operation is always performed in a well determined node (the inner-most substitution node). The adjunction operation, sketched in Figure 2, involves an initial/developing tree (*D-tree*$_{i-1}$) and an auxiliary tree (*A-tree*): it replaces the foot node of the auxiliary tree with the tree cropped down the adjunction node from the *D-tree*$_{i-1}$ and then it inserts the modified *A-tree* in the adjunction node, resulting thus a new developing tree (*D-tree*$_i$).

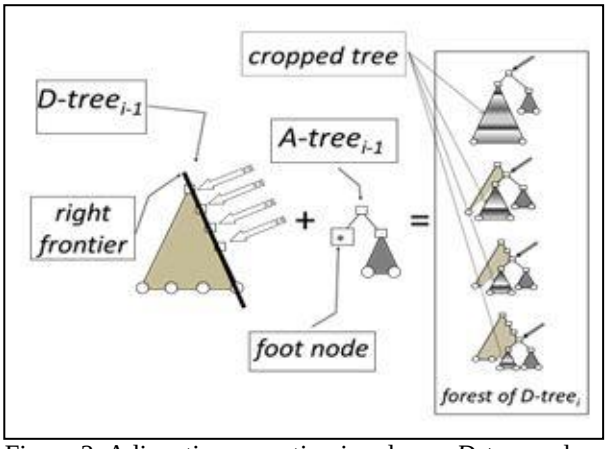

Figure 2: Adjunction operation involves a *D-tree* and an *A-tree* and producing a forest of *D-trees*

In Figure 2, the developing tree (*D-tree*$_{i-1}$) represents the discourse structure of the already analyzed text and the *A-tree* represents the discourse structure of the subtext processed in one step (an elementary discourse unit or a small discourse tree representing a sentence).

After each adjunction operation, potentially applied onto each node of the right frontier, a forest of developing trees is obtained. This leads to an exponential explosion of the developing structure, which can be mastered by ranking the trees on a global score associated to each tree, and using for the next step only the best placed ones (a kind of beam search).

In the research reported here we are preoccupied to cut down the computational complexity of the search during a VT-guided discourse parsing process. We show that at certain steps during the incremental process, when sufficient information exists, it is possible to keep open for adjunction only a subset of the right frontier, this way drastically reducing the explosion. This is done by focusing the adjunction on those nodes which maximize the chance to solve referential links on veins, by exploiting also details on the cropped and the auxiliary trees.

2.4 Centering on veins (VT score)

As shown in Section 2.1, VT suggests to associate scores to Centering transitions, this way becoming possible to quantify the coherence of a text. In Centering, transitions are computed on pairs of adjacent units within the borders of each segment of the discourse. VT argues that this computation can be generalized to the whole discourse by summing up the CT transitions' scores on *domains of referential accessibility* (DRA). The DRA of a unit *u* is given by the units in the vein expression of *u* that precede *u*, and the units are not necessarily adjacent any more. In Example 1, with its corresponding discourse tree structure represented in Figure 3, the centering score on veins is computed different from centering score proposed in (Grosz et al., 1995).

Example 1:

1. *As **John** came nearer,*
2. *he saw that the two men were his brothers,*
3. *who came from far away,*
4. *and **he** said 'I am happy to see you '.*

Centering is defined as a local theory of discourse structure, which makes it applicable only inside text segments. If the declared borders of CT would be forced and scores would be computed also over segment boundaries, no significant pairs' transitions scores would add to the overall score, because at most of segment boundaries no Cb's could be computed (thus, transition score equaling zero). This is the very definition of segment borders. As such, summing up the scores of all segments or totaling the overall score of the discourse as belonging to just one large segment would rather make no difference. If a unit has a predecessor in classical Centering, immediately to its left, in VT it is placed on a unit's DRA, therefore on the vein of some unit. Vein expressions, and hence DRAs, can skip segments' borders as defined in CT. As such, a text of *N* units in length adds on the overall score in VT the same number of transitions as in CT. Computation of CT scores could be extended to the whole discourse and a comparison could be drawn between the global extended_CT score and the global VT score. VT claims to be a global theory, because the segments limits are no more significant. Both theories

employ the same five transition types presented in Section 2.1, but VT claims that when considering transitions over veins they are consistently smoother.

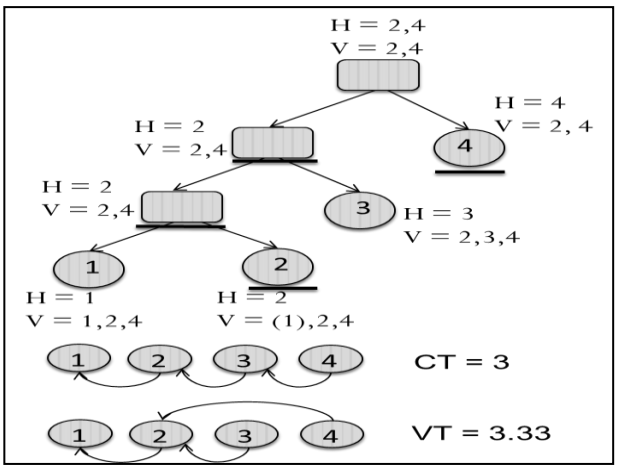

Figure 3: The CT and VT scores (H = head expressions, V = vein expressions)

The outlined nodes are nucleus and the others are satellites. As depicted in Figure 3, the CT score is computed in the following order: (1, 2), (2, 3), (3, 4):

Cf(U1) = {[John]};
Cb(U1) = [John];

Cf(U2) = {*he*=[John], [the two brothers]};
Cb(U2) = [John];
Cp(U2) = [John];
Transition (1, 2) = CONTINUING; score = 4;

Cf(U3) = {*who*=[the two brothers]};
Cb(U3) = [the two brothers];
Cp(U3) = [the two brothers];
Transition (2, 3) = SMOOTH SHIFTING; score = 2;

Cf(U4) = {*he=I*=[John], *you*=[the two brothers]};
Cb(U4) = [the two brothers];
Cp(U4) = [John];
Transition (3,4) = RETAINING; score = 3.

This leads to an average global CT score: (4+2+3)/3 = 3. The VT score is computed in a different manner, given by the vein expressions and DRA as described above. The transitions are: (1, 2), (2, 3), (2, 4). The difference from the previous computation consists in the last pair: (2, 4), instead of (3, 4). This will trigger a different Cb, because, now, "previous" with respect to unit 4 is unit 2 and not 3:

Cb(U4) = John;
Transition = CONTINUING; score = 4.

Thus, the average score is (4+2+4)/3 = 3.33, greater than in the CT case, indicating a smoother discourse.

3. The method

In the process of building discourse trees, a great importance is represented by the relationship between reference chains and the discourse structure (a manifestation of cohesion) on one hand and, on the other hand, between reference chains and the smoothness of centering transitions (a manifestation of coherence) (Cristea et al., 2005). We consider the Veins Theory a necessary step for observing the link between referential expressions from the incoming text segment and whole discourse.

A set of formulas deduced from VT rules helps to guess the most promising nodes of the right frontier where an adjunction can be made. The vein expressions of a node of the right frontier allows to predict how it will be changed in case an adjunction operation would be operated on it, knowing only the nuclearity of the adjoining node, the nuclearity configuration below the *A-tree* root node (*N_S*, *N_N* or *S_N*) and the referential chains that link the material node of the *A-tree* onto the previous discourse. With this information in hand, those nodes belonging to the right frontier which maximize a function of referentiality can be computed. This function counts the number of referential expressions belonging to the material node of the A-tree whose coreference chains intersect the vein expression after the adjunction on a certain node of the right frontier of the *D-tree*. Then, by taking the decision to adjoin the *A-tree* onto one of these points, we have adhered to a greedy strategy, assuming that the best choice made now will maximize the probability for the *D-tree* to further evolve onto the most cohesive and coherent structure. Using this information, the best nodes from right frontier where the adjunction should be made can be predicted. Thus, the search space for the attachment nodes is reduced, because only the nodes that contain referential expressions are targeted. Also, the coherence of the discourse structure is maximized, because the vein expression of the specific nodes will append the best fitting labels. This implies a maximization of the VT score.

3.1 Detailed description of the set of formulas

The set of formulas derives from VT and predicts what changes appear in vein expressions in the *D-tree* after an adjunction is made. If this is done without entirely computing a whole tree for each of the adjunction positions on the RF of the developing tree, then a lot of computations are saved. The prediction takes into consideration the nuclearity of the nodes below an adjunction node as well as the nodes below the root node of the *A-tree*. As after each adjunction, only few nodes of the developing tree are actually modified, it would be very good to maximize the combined score by comparing only the selected veins that differ from one tree to another of the adjunction forest. The optimum adjunction node on the RF can thus be chosen, reducing also the computation time.

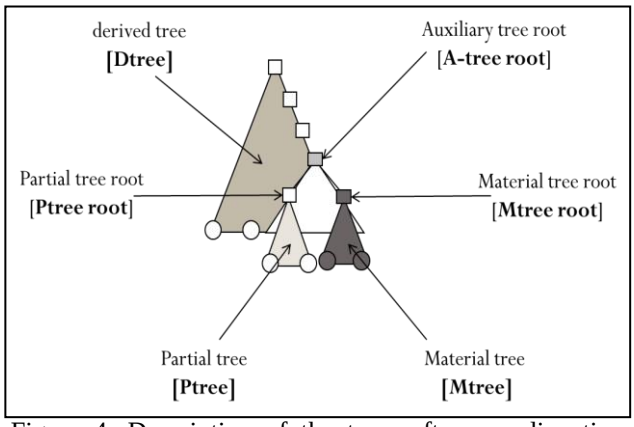

Figure 4: Description of the trees after an adjunction operation involved in the set of formulas

In Figure 4 the root of the Partial tree is the adjunction node belonging to the right frontier of the *D-tree* and the Material tree is the right children of the root of the *A-tree* (former right sibling of the foot node).

In the process of computing the formulas the following notations were taken into account:

- the vein expression of a node of P-tree is V_P ;
- the vein expression of a node of M-tree is V_M ;
- the vein expression of a node of D-tree is V_D ;
- the vein expression of P-tree root node is $V_{\mathbf{Pr}}$;
- the vein expression of M-tree root node is $V_{\mathbf{Mr}}$;
- the vein expression of A-tree root node is $V_{\mathbf{Ar}}$;
- the head expression of a node of P-tree is H_P ;
- the head expression of a node of M-tree is H_M ;
- the head expression of a node of D-tree: is H_D ;
- the head expression of P-tree root node is $H_{\mathbf{Pr}}$;
- the head expression of M-tree root node is $H_{\mathbf{Mr}}$;
- the head expression of A-tree root node is $H_{\mathbf{Ar}}$;

Analyzing how the resulted tree is changed, nine cases were discovered, determined by two factors: the presence of referential links between the incoming text and the initial discourse and the type[1] of A-tree root node (N_S, S_N, or N_N). They are presented below:

1) If the type of A-tree root node is N_S and adjunction is made in the root of the D-tree, then the following things happen:

- $V_{\mathbf{Ar}} = V_{\mathbf{Pr}}$;
- $H_{\mathbf{Ar}} = H_{\mathbf{Pr}}$;

2) If the type of the A-tree root node is N_S and adjunction is made in a nuclear node, then the following things happen:

- $V_{\mathbf{Ar}} = V_{\mathbf{Pr}}$;
- $H_{\mathbf{Ar}} = H_{\mathbf{Pr}}$;
- $V_M = seq(V_M, simpl(V_{\mathbf{Pr}}))$;

3) If the type of the A-tree root node is N_S and adjunction is made in a satellite node, then the following things happen:

- $V_{\mathbf{Ar}} = V_{\mathbf{Pr}}$;
- $H_{\mathbf{Ar}} = H_{\mathbf{Pr}}$;
- $V_M = seq(V_M, V_{\mathbf{Pr}})$

[1] By type, here and below, we mean the nuclearity configuration of children in the left to right order.

4) If the type of the A-tree root node is S_N and adjunction is made in the root of the D-tree, then the following changes happen:

- $V_M = seq(simpl(V_M), mark(H_{\mathbf{Pr}}))$.

If the update of the M-tree is done bottom-up on the right frontier then the process stops after the first satellite is met;

- $V_P = seq(V_P, V_{\mathbf{Ar}})$;
- $V_{\mathbf{Pr}} = seq(H_{\mathbf{Pr}}, H_{\mathbf{Ar}})$;

5) If the type of the A-tree root node is S_N and adjunction is made in a nuclear node, then the following things happen:

- The head of the P-tree root is marked $V_D = seq(V_D, H_{\mathbf{Ar}}, simpl(H_{\mathbf{Pr}}))$;

If the update of the D-tree is done bottom-up on the right frontier then the process stops after the first satellite is met;

$V_{\mathbf{Ar}} = seq(V_{\mathbf{Ar}}, seq(V_{\mathbf{Pr}}, simpl(H_{\mathbf{Pr}})))$;

- $V_P = seq(V_P, H_{\mathbf{Ar}})$;
- $V_M = seq(V_M, seq(mark(H_{\mathbf{Pr}}), V_{\mathbf{Pr}}))$;

6) If the type of the A-tree root node is S_N and adjunction is made in a satellite node, then the following things happen:

- $V_P = seq(V_P, V_{\mathbf{Ar}})$;
- $V_M = seq(V_M, seq((mark(H_{\mathbf{Pr}}), V_{\mathbf{Pr}}))$;
- The head of the P-tree root is marked $V_{\mathbf{Ar}} = seq(V_{\mathbf{Ar}}, seq(V_{\mathbf{Pr}}, simpl(H_{\mathbf{Pr}})))$;

7) If the type of the A-tree root node is N_N and adjunction is made in the root node, then the following things happen:

- The new P-tree will be the previous D-tree and it will copy all previous vein and head expressions;
- $V_M = seq(V_M, V_{\mathbf{Pr}})$;
- $V_P = seq(V_P, V_{\mathbf{Ar}})$;

8) If the type of the A-tree root node is N_N and adjunction is made in a nuclear node, then the following things happen:

- $H_{\mathbf{Ar}} = seq(H_{\mathbf{Ar}}, H_{\mathbf{Pr}})$;
- $V_{\mathbf{Ar}} = seq(V_{\mathbf{Ar}}, V_{\mathbf{Pr}})$;
- $V_D = seq(V_D, V_{\mathbf{Ar}})$;

If the update of the D-tree is done bottom-up on the right frontier then the process stops after the first satellite is met;

- $V_P = seq(V_P, V_{\mathbf{Ar}})$;

9) If the type of A-tree root node is N_N and adjunction is made in a satellite node, then the following things happen

- $H_{\mathbf{Ar}} = seq(H_{\mathbf{Ar}}, H_{\mathbf{Pr}})$;
- $V_{\mathbf{Ar}} = seq(V_{\mathbf{Ar}}, V_{\mathbf{Pr}})$;
- $V_P = seq(V_P, V_{\mathbf{Ar}})$;

3.2 Exemplifying how the set of formulas are used

To understand how the set of formulas are applied, let's consider the text of Example 2, already segmented in seven units:

Example 2:

1. *Makaha changes its Name.*
2. *Makaha Inc. said:*
3. *the **CEO** has decided that the new name will be TerroCom*
4. *In a new release, the company said*
5. *the new name more accurately reflects focus on high-technology communications,*
6. *including business and entertainment software, interactive media and wireless data and voice transmission.*
7. ***He** decided to make this change starting with tomorrow.*

The demonstration that follows is built on the supposition that the first six clauses of the text in Example 2 are already analyzed and the incremental process has reached the point where unit no. 7 has to be adjoined to the right frontier of the developing tree. Let's note that this adjoining operation will trigger modifications of the vein expressions of some (or all) of the nodes of the terminal frontier. This suggests the idea that is put at the core of our proposal: find that node of the right frontier where the adjunction of a new material node containing the current unit will prolong the vein expressions of the terminal nodes in the most profitable way.
The first step is to identify which units of the previous discourse contain antecedents for the anaphors contained in the current node. For instance, in Example 2, we want to add label 3 in the vein expression of unit 7 because both units 3 and 7 include references to the same entity. This way the coherence of the text is kept high, since on the argumentation line of unit 7 there will be a transition scored high (most probably CONTINUATION or RETAINING, conforming to Centering (Grosz et al., 1995) and VT (Cristea et al., 1998a).

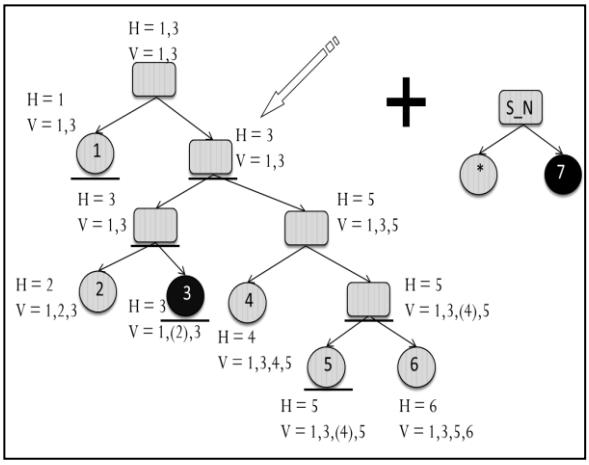

Figure 5: Discourse tree representation of the text in Example 2

In Figure 5 the arrow points to the node where the adjunction must be made. Head (H) and vein (V) expressions are marked on each node. The selection was made because the attachment node is situated on the *right frontier*, it covers the node representing clause *3* and is the nearest to it. The tree resulted after adjunction is presented in Figure 6. As can be seen, the label of node 3 is placed on the vein expression of the node representing unit 7. A set of formulas derived from VT proves that if the A-tree root node is labeled N_N and the adjoining node is nuclear (N), all the vein expressions contained in the D-tree will be kept unchanged but in the same time, new information is added. This is easily observed from the set of vein expressions. Thus, when the VT score is computed from this example, it will take into account the transition between units (3, 7), which is also what we intended.

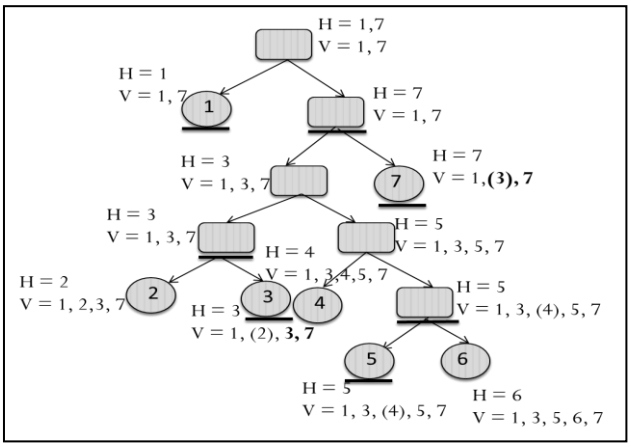

Figure 6: Resulted discourse tree after adjunction operation

Based on the set of formulas and analyzing how the resulting tree is changed, the selection for the best node of the RF where the adjunction takes place can be made by considering these cases:
There is no referential link between a unit in the *D-tree* and a unit in the *A-tree*:

- If the *A-tree* root node is typed *S_N* or *N_N*: adjunction should be made in a *satellite* (*S*) node (an adjunction in a *nucleus* (*N*) would encumber the deletion of the head expression of the cropped tree);
- If the *A-tree* root node is typed *N_S*: do adjunction in either an *S* or an *N* (better in *S* because it keeps the marked nodes).

There is at least one referential link between a unit in the *D-tree* and a unit in the *A-tree*:

- If the *A-tree* root node is typed *S_N* or *N_N*: do adjunction in a *nuclear* node, which will cover the whole referents (the nuclear node for adjunction must be the one who contains most of the referential expressions);
- If the *A-tree* root node is typed *N_S*: do adjunction in a *satellite* node (the marked nodes will not be lost and therefore the coherence of text is preserved though maintaining the referential expressions).

4. Evaluation

To evaluate our method we used the English part of the corpus mentioned in the original VT paper (Cristea et al., 1998b), texts distributed by the Message Understanding Conference (MUC-7). This corpus includes 30 newspaper texts whose lengths varies widely (average of 408 words and standard deviation of 376 words) and are manually annotated for co-reference relations (Hirschman and Chinchor, 1997) and complemented with RST structure, by Marcu et al., (1999). In Table 1 we present the results obtained comparing two different methods. The first one explores all the nodes belonging to the *right frontier* and the second one uses the method presented in this paper.
Comparisons were made taking into consideration two factors:

- the coherence of the discourse trees;
- the structure of the discourse trees.

The coherence of the discourse tree was evaluated using the method proposed in Section 2.4. The discourse tree structures were compared using the measures proposed in Mitocariu et al. (2013). The significance of the labels in the table is the following:

- ADJ- : the number of adjunction operations with the optimization feature switched off;
- ADJ+ : the number of adjunction operations with the optimization feature switched on;
- CT : the classical Centering score;
- VT- : the Centering score on veins with the optimization feature switched off;
- VT+ : the Centering score on veins with the optimization feature switched on;
- OS, NS, VS: scores for comparing discourse tree structures in terms of coverage, nuclearity and vein expressions (Mitocariu et al., 2013).

Filename	ADJ-	ADJ+	CT	VT-	VT+	OS	NS	VS
MUC1	169	83	0.121	**0.121**	**0.121**	0.939	0.909	0.972
MUC2	27	15	0.286	**0.429**	**0.429**	1	1	1
MUC5	21	12	0	**0**	**0**	1	1	1
MUC7	33	18	0	**0**	**0**	1	1	1
MUC9	57	30	0.304	**0.348**	**0.348**	1	1	1
MUC10	51	27	0.091	**0.091**	**0.091**	1	1	1
MUC11	171	84	0.096	**0.096**	**0.115**	0.962	0.942	0.973
MUC13	21	12	0.333	**0.5**	**0.5**	1	1	1
MUCl4	75	39	0.1	**0.1**	**0.1**	1	1	1
MUC16	69	36	0.588	**0.647**	**0.647**	1	1	1
MUC17	9	6	0.6	**0.6**	**0.6**	1	1	1
MUC18	30	15	0.667	**0.75**	**0.667**	0.917	0.917	0.984
MUC20	27	15	0	**0**	**0**	1	1	1
MUC22	273	132	0.11	**0.11**	**0.134**	0.951	0.927	0.962

Table 1: Comparing the two approaches.

The figures in the table above show that applying the set of formulas, the results of the discourse parsing system are similar or almost identical with the system running without applying the set of formulas. The results are important because they show a significant reduction of the computational effort.

Analyzing the fields ADJ- and ADJ+, it can be noticed that the number of adjunction operations in classical incremental discourse parsing is larger than the number of adjunction operations performed when applying the set of formulas. From these results is easy to conclude that the execution time is reduced approximately by half. For example, for the text MUC1 the number of operations for ADJ- is 169 and for ADJ+ are 83. Decreasing the number of operations triggers the reduction of the execution time. The second important finding is that the structures obtained by applying the reduction strategy have a similar quality as those obtained using the classical incremental parsing. Almost identical discourse structures are obtained with the feature switched on as with it switched off (most of the comparison scores are 1 or very close to 1). Thus, the economy in running effort does not negatively affect the coherence of the obtained structures.

5. Conclusions

We have proposed a set of formulas which may be used by incremental discourse parsing systems to reduce the number of adjunction operations on the right frontier and we demonstrated how these formulas help to maintain the coherence of the text and to reduce the complexity of the computations. The incremental evaluation of tree structures is based on Veins Theory. We make use of referential expressions, the nuclearity of the adjunction nodes and the type of the auxiliary tree to select the best node to make the adjunction operation. From the perspective of an incremental discourse parsing system, this set of formulas is useful in processing long texts, where it helps to reduce the search space of adjoining on the *right frontier*.

6. References

Stergos D. Afantenos and Nicholas Asher. (2010). Testing SDRT's Right Frontier. *In Proceedings of COLING*, pp. 1–9.

Daniel A. Anechitei, Dan Cristea, Ioannidis Dimosthenis, Eugen Ignat, Diman Karagiozov, Svetla Koeva, Mateusz Kopeć and Cristina Vertan. (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.

Nicholas Asher. (1993). Reference to Abstract Objects in Discourse. Dordrecht: *Kluwer Academic Publishers*.

Nicholas Asher and Alex Lascarides. (2003). Logics of Conversation. *Cambridge University Press*.

Branimir K. Boguraev and Mary S. Neff. (2000). Lexical cohesion, discourse segmentation and document summarization. *In Proceedings of RIAO*.

Dan Cristea, Oana Postolache, Ionut Pistol. (2005). Summarisation through Discourse Structure. *In the 6th International Conference CICLing*, Mexico City, Mexico, pp. 632–644.

Dan Cristea. (2005). The Right Frontier Constraint Holds Unconditionally. *In Proceedings of the Multidisciplinary Approaches to Discourse (MAD'05)*,

Chorin/Berlin, Germany.

Dan Cristea, Nancy Ide, Laurent Romary. (1998a). Veins theory: A model of global discourse cohesion and coherence. *In Proceedings of the 17th international conference on Computational linguistics*, pp. 281-285, Montreal.

Dan Cristea, Nancy Ide, Laurent Romary. (1998b). Marking-up Multiple Views Of A Text: Discourse And Reference. *In Proceedings of the First International Conference on Language Resources and Evaluation*, Granada.

Dan Cristea and Bonnie L. Webber. (1997). Expectations in Incremental Discourse Processing. In Philip R. Cohen, Wolfgang Wahlster (eds.) *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, Madrid.

Laurence Danlos. (2008). Strong generative capacity of RST, SDRT and discourse dependency DAGs. In Benz, A. and P. Kühnlein, editors, *Constraints in Discourse, Pragmatics and Beyond New Series*, pp. 69–95.

Barbara J. Grosz, Aravind K. Joshi and Scott Weinstein. (1995). Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2), pp. 203–226.

Lynette Hirschman and Nancy Chinchor. (1997). Muc-7 Coreference task definition. *In MUC-7 Proceedings. Science Applications International Corporation*.

Aravind K. Joshi and Yves Schabes. (1997). Tree-Adjoining Grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal languages*, pp. 69-123, Springer, Berlin.

William C. Mann and Sandra A. Thompson. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *TEXT*, 8(3), pp. 243-281.

Daniel Marcu. (2000). The Theory and Practice of Discourse Parsing and Summarization. *The MIT Press*. Cambridge, Massachusetts.

Daniel Marcu, Estibaliz Amorrortu and Magdalena Romera. (1999). Experiments in Constructing Discourse Trees. A corpus of discourse trees. *In Proceedings of the ACL Workshop on Standards and Tools for Discourse Tagging*, College Park, MD, pp. 48-57.

Elena Mitocariu, Daniel A. Anechitei, Dan Cristea. (2013). Comparing discourse tree structure. *In the 14th International Conference CICLing*, Samos, Greece, March 24-30, pp. 513-522.

Yoshio Nakao. (2000). An algorithm for one-page summarization of a long text based on thematic hierarchy detection. *In Proceedings of the ACL*, pp. 302-309.

Bonnie L. Webber. (1991). Structure and ostension in the interpretation of discourse deixis. *Natural Language and Cognitive Processes*, 6(2):107–135.

Masao Utiyama and Hitoshi Isahara. (2001). A statistical model for domain-independent text segmentation. *In Proceedings of the ACL*, pp. 499-506.