Summarizing documents based on cue-phrases and references

Dan Cristea
„Al.I.Cuza“ University of Iasi
Faculty of Computer Science
and
Romanian Academy
Institute of Theoretical Computer Science – the Iasi branch
dcristea@infoiasi.ro

Oana Postolache, Georgiana Puşcaşu, Laurenţiu Ghetu
„Al.I.Cuza“ University of Iasi
Faculty of Computer Science
16, Berthelot St.
6600 – Iasi, Romania
{oanap, georgie, laug}@infoiasi.ro

Abstract

The paper presents a method of building the discourse structure of a discourse by combining indications on local structure given by cue-phrases with indications on global structure given by references found in the text. Then the discourse structure is used to obtain focused summaries.

1 Introduction

There is generally accepted that a strong correlation exists between the structure of discourse and referentiality (Fox, 1987; Vonk *et al*., 1992, Cristea *et al*., 2000). On another hand, document summarization could take advantage from knowing the structure. Marcu (2000), for instance, has shown how parameterized summaries of a document can be build provided its rhetorical structure is known. Putting all together, one can arrive, without any surprise, at the strong interdependence between referentiality and summaries, intermediated by the discourse structure. A simple argument in support of this interdependence is that a summary cannot be coherent if it contains dangling references.

In this paper we present a method to obtain coherent focused summaries based on the discourse structure of the discourse, which is partially inferred from indications on local structure given by cue-phrases and partially from references found in the text.

The summaries that we obtain are extracts (Mani, 2001). We say that a summary of a document is focused on a certain discourse entity if the summary reveals on short what the document tells about the key-entity, within the context of the whole document. A possible scenario addressing the need for a focused summary is that of a user interested in reviewing scientific texts, in particular in findings on a certain drug. Using Google or another search engine she gets a tremendous lists of documents mentioning the searched entity. Since time does not allow her to read all the found documents, abstracts would be of value. The problem with a general abstract that can be obtained by passing the task to a common abstracting engine is that the item searched could be secondary to the theme of the document, in which case it will not be included in the generated abstract. The user would be interested to know, briefly, why is that entity mentioned in a document, therefore a focused abstract.

At the base of our method stays the assumption that if summarization is the goal, a less precise discourse structure is sufficient. To obtain it, cue-words and phrases are good indicators of local structural interdependencies between elementary discourse units (*edu*s); based on cue-phrases, elementary sentence-level trees (*sdt*s) are inferred; they are then integrated into a global coherent discourse tree using indications on discourse structure brought by references, as outlined by veins theory (Cristea *et al*, 1998).

The paper is structured as follows: section 2 presents the method, sections 3 and 4 present the basics (veins theory and the resolution of anaphora), section 5 describes a set of consistency constrains for the discovery of *sdt*s, sections 6 displays the methods of integration of the *sdt*s into a whole discourse tree based on references, section 7 presents the data employed in the experiment, and some results, and section 8 discusses possible extensions.

2 The method

A text can be read in many ways. Practically each *edu* of the text gives a specific perspective from which to interpret the whole text. Such a perspective centred on a certain *edu* should be thought as revealing what would be the meaning of that particular *edu* in the overall context. Cristea *et al* (1998) propose a theory to evidence centred interpretations, cut up from a rhetorical-like discourse tree structure. The vein expressions defined there constitute means to look at *edu*s from inside out. Each such vein expression, claiming to express what the text says about that specific *edu* in the overall context, gives also a way of summarising the text, focussed on the entities mentioned in that *edu*.

Many discourse parsing methods have been described (Cole *et al*., 1995; Marcu, 2000; Cristea, 2000). Cristea (2000), for instance, presents an incremental discourse parsing method, which places at the base the principle that a text has the one discourse structure, that displays the smoothest centering transitions (Grosz *et al*., 1995) along veins, as well as most of the references satisfied along the veins. These two criteria combine to score a number of plausible partial trees and to retain at each step of the incremental development N best scored trees. Unlike Cristea (2000), the method proposed here does not first build a tree in order to accept it, if well scored, or to filter it out, if badly scored, but rather uses references as a guide during the development the tree.

The other clue used in building the structure is given by cue-phrases (Knott and Dale, 1992), (Marcu, 2000) which are used to build *sdt*s. To do that, we use Soricutu and Marcu's (2003) claim that the text span corresponding to one sentence is merely covered by one node in the structure (more than 90% of the cases, according to them).

The preparatory phases suppose POS-tagging (Tufis, 1999), syntactic tagging done by an FDG parser and NP-tagging (Ait-Mohtar, and Chanod, 1997). Then *edu* are detected (Puscasu, forthcoming) based on the identification of finite verbs and detection of their syntactic roles. Local corrections, mainly due to cue-phrases, are also possible.

Following, *sdt*s are build (as will be described in section 5). In parallel, or following the *sdt*-building phase, antecedents of anaphors are looked for, by running the AR-engine (described in section 4). The chains of co-referential links are then used to sew pieces together in a complete discourse structure (described in section 6).

Having the discourse structure, the vein expressions of the *edu* containing the entity search for will configure the output summary.

3 Veins theory

By using the RST notion of nuclearity, veins theory (VT) (Cristea *et al*., 1998), (Ide&Cristea, 2000) reveals a "hidden" structure in the discourse tree, called *vein*, which enables to evidence for each unit of a discourse a *domain of evocative accessibility* (*dea*) as a string of units where antecedents of anaphors belonging to the unit should be found.

The fundamental intuition underlying the unified view on discourse structure and accessibility in VT is that the RST-specific distinction between nuclei and satellites constrains 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 for each anaphor a *dea*.

The observations that underline the computation of vein expressions in VT are as follows (discourse units are noted here after u_1, u_2, u_3, while relations R, R_1, R_2; when used as arguments of relations, the units' nuclearity will be marked by a superscript ^{n} – for nucleus, and ^{s} – for satellite; we will say that "a unit u_2 refers a unit u_1" and we will understand "a referential expression belonging to a unit u_2 refers a discourse entity also referred from the unit u_1"):

- a satellite or a nucleus can refer a nuclear sibling to its left: in sequences $u_1^n\ R\ u_2^s$, or $u_1^n\ R\ u_2^n$, u_2 can refer u_1;
- a nucleus can refer its own left satellite: in sequences $u_1^s\ R\ u_2^n$, u_2 can refer u_1;
- a right satellite of a nucleus u cannot be accessed from another right sibling, nuclear or satellite, of u: in sequences $(u_1^n\ R_1\ u_2^s)^n\ R_2\ u_3^n$ or $(u_1^n\ R_1\ u_2^s)^n\ R_2\ u_3^s$, u_3 can refer u_1 but not u_2;
- a nucleus blocks the accessibility from a right satellite to a left satellite: in sequences $(u_1^s\ R_1\ u_2^n)^n\ R_2\ u_3^s$, u_3 can refer u_2 but not u_1.

VT contributes with a view on top-down summarization, similar to Marcu's (2000), while also revealing how focused summaries can be produced.

4 Anaphora Resolution

In (Cristea and Dima 2001), (Cristea *et al.*, 2002a) a framework incorporating a general anaphora resolution (AR) engine and able to accommodate different AR models is proposed. This approach sees the linguistic and semantic entities involved in the cognitive process of anaphora resolution represented on three layers: the **text layer** – populated with referential expressions (*re*s), the **projected layer** – where feature structures are filled-in with information fetched from the text layer (in the following, projected structures – *ps*s) and the deep **semantic layer** – where discourse entities (*de*s), actually a representation of the entities the discourse talks about, are placed. It is said that a *ps* is **projected** from an *re* and that a *de* is **proposed** or **evoked** by a *ps*.

Within the AR-engine framework an AR model is defined in terms of four components: a set of attributes and the corresponding types of the objects populating the projection and semantic layers, a **set of knowledge sources** (virtual processors) intended to fetch values from the text to the attributes of the *ps*, **a set of matching rules and heuristics** responsible to decide whether the *ps* corresponding to an *re* introduces a new *de* or, if not, which of the existing *de*s it evokes, and **a set of heuristics that configure the domain of referential accessibility**, establishing the order in which *de*s have to be checked, or certain proximity restrictions.

In (Cristea *et al*., 2002a), pronominal as well as noun anaphora were investigated. To a great extend, the results proved the initial hypothesis, namely that models behave better and better as more features are fired. For a small corpus of about two pages taken from the novel "1984" of G. Orwell (where five characters have been tracked, whose co-reference chains in the golden annotation had lengths of 23, 14, 3, 25 and 16 referential expressions), the best models experienced proved 100% precision and a recall in the range 70% to 100%. In another research (Cristea *et al*., 2002b) the investigation was extended over cases generally considered difficult to tackle (co-reference resolution triggered by positional constrains, common nouns anaphor and antecedent with disagreement in lemma, noun and pronoun anaphors displaying number disagreement with the antecedents, bridging anaphora, as well as anaphoric references other than net co-references).

5 Consistency constraints for elementary discourse trees

In this section we propose a representation and a method of determining safe inter-*edu*s local dependencies, contributed by cue words or phrases (in the following, called *markers*). The dependencies will configure an elementary discourse tree structure covering mainly a sentence (sometimes even more than that) in which inner nodes are labelled with markers and terminal nodes with *edu* labels. Each node of the tree is also marked by a nuclearity function in the set {*n*, *s*} (for nuclear, satellite) such that at each level, between the two descendents of an inner node, at least one is marked *n*.

Example 1:

[*John is determined to pass the NLP exam*[1]] so, because [*he has missed many courses* [2]] and [*was only vaguely implicated at the working sessions* [3]] . [*he will have a hard time until summer.* [4]]

In this example, the notation indicates the segmentation in *edu*s (in square brackets) and the cue words with an impact in the determination of the discourse structure (underlined): *so* – indicates that a unit subordinate to the preceding one follows; *because* (following another cue word, as well as in a sentence-initial position) – indicates that the unit it is prefixing is a subordinate of a unit that follows after this one; *and* – indicates a conjunction between two units of equal nuclearity that prefix and, respectively, succeeds it. This arrives at associating to markers argument patterns, as suggested in Figure 1:

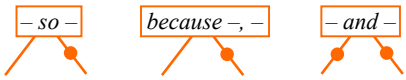

Figure 1: Arguments patterns of cue-phrases

Example 1 displays the following distribution of *edu*s and markers: 1 *so, because* 2 *and* 3, 4.

On each branch stemming out of a marker in Figure 1, virtually any domain of arguments, obtained by a combination of the units laying in the text on the corresponding part, could be formed. In the following notation, we will mark these domains as ordered lists of discourse units, while also underlying the nuclear arguments:

[1] *so* [2,3,4]
because [2,3,4], [2,3,4]
[1,2] *and* [3,4]

Among all possible combinations of lists of *edu*s encumbered by this scheme, we will reject from the beginning the empty lists, as well as those whose content concatenation display gaps in the sequence. The following pairs of argument-lists remain:

– so –	*because –, –*	*– and –*
[1] – [2,3,4]	[2] – [3,4]	[1,2] – [3,4]
[1] – [2,3]	[2,3] – [4]	[2] – [3,4]
[1] – [2]		[1,2] – [3]
		[2] – [3]

Among the 3 * 2 * 4 = 24 possibilities, many are still inconsistent. The following rules state further constrains (we will note the lists with M_1, M_2, etc.). They express natural conditions of tree well-formedness:

The "nesting-arguments" rule:

If $x \in M_i \cap M_j$ with $i \neq j$, then either $M_i \subseteq M_j$ or $M_i \supseteq M_j$.

This rule states that it is impossible to have two inner nodes of the tree, which cover crossing text spans on the terminal frontier.

The combination M_1=[1], M_2=[2,3,4] (for *so*), M_3=[2], M_4=[3,4] (for *because*), and M_5=[1,2], M_6=[3,4] (for *and*) do not abbey this rule, because $2 \in M_2$, $2 \in M_5$ and neither $M_2 \subseteq M_5$, nor $M_2 \supseteq M_5$.

Instead, the combination M_1=[1], M_2=[2,3,4] (for *so*), M_3=[2], M_4=[3,4] (for *because*), and M_5=[2], M_6=[3,4] (for *and*) do abbey the nesting arguments rule.

The "balanced-displacement" rule:

For any two *edu*s x, y placed in sequence (x before y), at least a marker, denoted by m, exists such that: $x \in$ left_subtree(m) and $y \in$ right_subtree(m).

This rule forbids the existence of dangling *edu*s in an elementary discourse tree. It stems from the assumption that a sufficient number of markers are found in the text. There where the text contributes with no marker, an empty cue-word ∅ is considered instead, with the default argument pattern: – ∅ –. Any of the combinations of nuclearity labels (n, n), (n, s), (s, n) are possible for its arguments.

In the example above, the combination of lists M_1=[1], M_2=[2,3,4] (for *so*), M_3=[2], M_4=[3,4] (for *because*), and M_5=[2], M_6=[3,4] (for *and*) do not obey this rule, because for *edu*s 3 and 4 there is no marker with 3 in its left sub-tree and 4 in its right sub-tree.

The "unique-root" rule:

There is one and only one marker that covers the sequence of all *edu*s.

In the example above, the combination of lists M_1=[1], M_2=[2,3,4] (for *so*), M_3=[2], M_4=[3,4] (for *because*), and M_5=[1,2], M_6=[3,4] (for *and*) do not obey this rule, because both *so* and *and* do cover the whole range of *edu*s.

The "one-parent" rule:

There are no two lists $M_i = M_j$ with $i \neq j$.

This rule asserts the obvious condition in trees that is impossible to have one text span which is an argument to two distinct markers.

For instance, the combination M_1=[1], M_2=[2,3,4] (for *so*), M_3=[2], M_4=[3,4] (for *because*), and M_5=[2], M_6=[3,4] (for *and*) contradicts twice this rule because of the lists $M_3=M_5$ and $M_4=M_6$.

Among the 24 possible combinations of lists of the above example, only one obeys all four rules: M_1=[1], M_2=[2,3,4] (for *so*), M_3=[2,3], M_4=[4] (for *because*), and M_5=[2], M_6=[3] (for *and*), which is also the expected one, displaying the sentence-level tree of Figure 2.

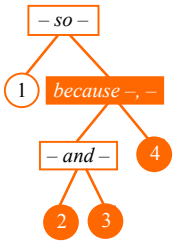

Figure 2: The elementary discourse tree of Example 1 (nuclear nodes filled-in)

The above rules applied to sentence-initial markers yields the integration of adjacent sentences into larger elementary tree.

6 The processing model

This section describes how pieces can be sewed together. The idea is to integrate elementary trees into a global one by taking into consideration references discovered by the AR-engine and arranging the *edu*s the references belong to such that they lay mostly along unit's vein expression.

Processing follows the following three phases:

1. determination of co-referential chains;
2. building sentence level discourse trees (*sdt*s) based on intra-sentence markers;

3. integration of *sdt*s up to a global discourse tree.

During the first phase, AR-engine is run over the POS-tagged, FDG-analysed, and NP-tagged text, as explained in section 4. The result is a set of co-referential chains of *re*s. All *re*s belonging to each such chain point to a unique *de*.

During the second phase, the syntactic constrains encumbered by cue-phrases at (mainly) sentence level are applied, in order to arrive to a sequence of *sdt*s, as explained in section 5. Then, during the third phase the sentence level trees are combined in sequence with the aim to obtain one complete tree of the whole discourse.

Let's note that the model we describe is opened for both an incremental as well as a pipe-line type of processing. In incremental processing, suppose a partial discourse tree (*pdt*) is obtained from processing the text up to (and excluding) the current *edu*. The current sentence *s* is submitted to the first phase, as described above, resulting in a set of co-reference relations from all anaphors contained in *s*. Then, as a result of running the second phase, an *sdt t*, is obtained from *s*. Finally, the third phase will integrate *t* into *pdt*, leaving a larger *pdt*. In a pipe-line type of processing, the first phase is run over the whole document, leaving a set co-referential expressions. In parallel with this phase or following it, the second phase will be run over the whole text, leaving a sequence of *sdt*s. Finally, the third phase will be run, in order to integrate the sequence of *sdt*s into a global discourse tree by taking into consideration the set of co-references.

The following discussion applies to both processing models. We will suppose the parser is in a state when the first *i sdt*s have been combined into a partial discourse tree structure, pdt_i, and the next *sdt* under operation is sdt_{i+1}. This tree has to be combined with the developing tree pdt_i by adjoining an auxiliary tree obtained from this one on the right frontier of the developing tree (Cristea&Webber, 1997). An auxiliary tree of an *sdt t* consists of a relation node, with a dummy (foot) node as its left child and *t* as its right child. Figure 3 displays the adjoining operation. If the name of the relation rooting the auxiliary tree is ignored, still two other problems are to be dealt with: to what node of the right frontier of the developing tree should the adjunction be directed, and what should be the nuclearity pattern of the two descendents of the relation node in the auxiliary tree (the foot node and sdt_{i+1})?

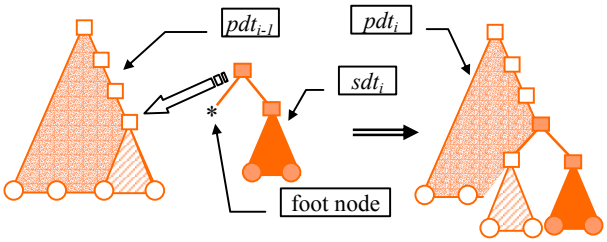

Figure 3: The adjoining operation

We define a function that records the number of co-references between *edu*s belonging to different *sdt*s, as follows.

If SDT^D is the ordered set of all *sdt*s over discourse *D* (indexed from left to right, as the text unfolds), and U^D is the set of all *edu*s in *D*, then:

f: SDT^D x $P(U^D)$ x SDT^D x $P(U^D) \rightarrow N$

(where, if *x* is a set, $P(x)$ is the power set of *x*, and N is the set of natural numbers), defined as follows: if u_k is an *edu* on the terminal frontier of *sdt* t_i, and u_l is an *edu* on the terminal frontier of *sdt* t_j (* represents all *edu*s on the terminal frontier of $SDT\ t_i$, respectively $SDT\ t_j$) then:

$f(t_i, u_k, t_j, u_l)$ = number of antecedents belonging to unit u_k of t_i directly or indirectly referred by anaphors belonging to unit u_l of t_j;

$f(t_i, *, t_j, u_l)$ = number of antecedents belonging to all units of t_i directly or indirectly referred by anaphors belonging to unit u_l of t_j (if two or more anaphors in u_l refer the same antecedent belonging to t_i then *f* will count all of them);

$f(t_i, u_k, t_j, *)$ = number of antecedents belonging to the unit u_k of t_i directly or indirectly referred by anaphors belonging to all units of t_j;

$f(t_i, *, t_j, *)$ = number of antecedents belonging to the all units of t_i directly or indirectly referred by anaphors belonging to all units of t_j;

$f(t_i, u_k, t_j, head(root(t_j)))$ = number of antecedents belonging to the unit u_k of t_i directly or indirectly referred by anaphors belonging to those units of t_j that are contained in the head expression of the root of t_j.

The following rules give decision criteria with respect to the node of the right frontier of the developing tree pdt_{i-1} where adjoining is to be operated:

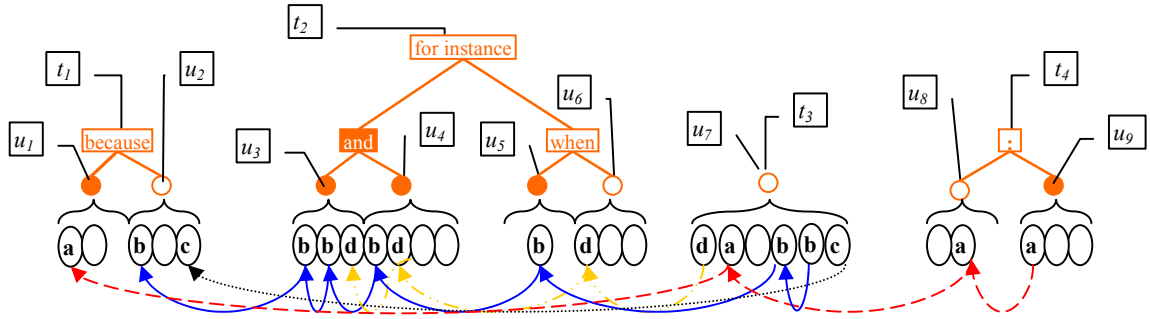

Figure 4: *Sdt*s and references for Example 2
(a = *Maria*, b = *Simon*, c = *the child*, d = *I*, empty = any other REs)

Rule 1:

If $f(t_p, u_k, t_i, *) > f(t_q, u_l, t_i, *)$, where $1 \leq p \neq q < i$ (meaning: the elementary tree t_i comes in sequence after the elementary trees t_p and t_q), u_k is any unit of t_p and u_l is any unit of t_q, then if n_k is the right frontier node of pdt_{i-1} (that contains both t_p and t_q) covering u_k or the lowest node on the right frontier of t_p that contains u_k in its head expression, then the auxiliary tree stemmed out of t_i is adjoined onto the node n_k.

Rule 2:

If $f(t_p, u_k, t_i, *) < f(t_q, u_l, t_i, *)$, with $1 \leq p \neq q < i$, but u_l is not visible on the right frontier of pdt_{i-1}, then u_l is ignored.

Rule 3:

If $f(t_p, u_k, t_i, *) = f(t_p, u_l, t_i, *)$, with $1 \leq p \neq q < i$ and $l < k$, then u_l is ignored.

Rule 4:

If $f(t_p, u_k, t_i, *) = f(t_q, u_l, t_i, *)$, with $1 \leq p \neq q < i$, but $f(t_p, u_k, t_i, head(root(t_i))) < f(t_q, u_l, t_i, head(root(t_i)))$ then t_i is adjoined into the node n_l, even if $p > q$;

Rule 5:

If there is no $p < i$ such that $f(t_p, u_k, t_i, *) > 0$, with u_k any unit of t_p, or if such a p exists but none of its u_k are visible on the right frontier of pdt_{i-1}, then t_i is adjoined onto the lowest most node of the right frontier of pdt_{i-1} as a satellite of it.

The root of the auxiliary tree being adjoined always remains with the nuclearity of the node where the adjoining is being made. What still has to be decided is the nuclearity of the foot node (which will give the nuclearity of the node onto which the adjunction is being made, let's call it u_k) and of its sibling (the current *sdt* t_i):

Rule 6:

The node u_k, where the adjoining is being made will always be nuclear.

Rule 7:

If $f(t_p, head(root(u_k)), t_i, *) > 0$ then t_i will be nuclear, otherwise it will be satellite.

We will display how the model works on the following example:

Example 2

[Maria went alone to the market [1]*] because [Simon had to stay at home with the baby.* [2]*] [Simon is a good friend of mine* [3]*] and [he also helped me in a number of situations.* [4]*] For instance [he was very helpful* [5]*] when [I had the problem with the car.* [6]*] [I think she has a lot of trust in him to let him alone with the child.* [7]*] [You know how Maria is* [8]*] : [she is not very hurried to give credit to anybody.* [9]*]*

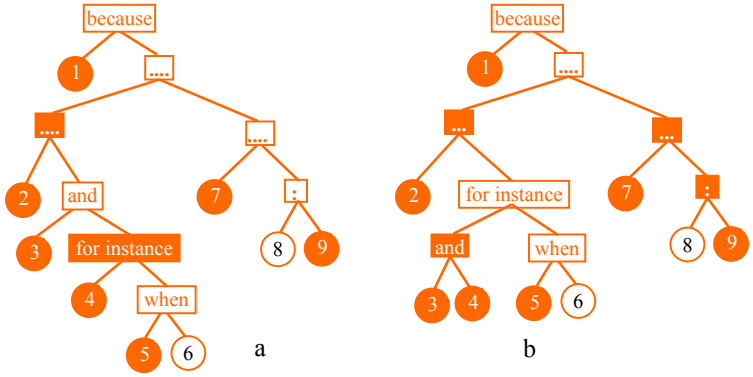

Figure 5: Correct (a) and computed (b) tree

Figure 4 shows the sequence of *sdt*s and the co-reference chains after the first two phases (in a pipe-line processing). The four *sdt*s are grouped together by three adjoining operations. At each step i, the function $f(t_p, u_k, t_{i+1}, *)$ is computed for each $p < i$, and any u_k belonging to t_p, then the rules described above are applied. Only the first step is detailed here: the *sdt* t_2 has to be adjoined onto the first *pdt*, which is t_1: $f(t_1, u_1, t_2, *) = 0$; $f(t_1, u_2, t_2, *) = 3$; (corresponding to the 3 references for *Simon* in u_2, from u_3 and u_4). Applying rule 1 t_2 must be adjoined onto t_1 at the level of u_2. According to rule 6, u_2 will be nuclear, while t_2 will be satellite,

according to rule 7 (the root of u_2 is the relation node `because`, whose head expression is u_1, and $f(t_1, u_1, t_2, *) = 0$). Figure 5 shows the correct tree, drawn by hand, compared with the computed one. The table below displays the vein expressions of the correct tree compared with the computed one.

	golden	computed
1	1	1
2	1 2	1 2 7 9
3	1 2 3 4	1 2 3 4 7 9
4	1 2 3 4	1 2 3 4 7 9
5	1 2 3 4 5	1 2 3 4 5 7 9
6	1 2 3 4 5 6	1 2 3 4 5 6 7 9
7	1 2 7	1 2 7 9
8	1 2 7 8 9	1 2 7 8 9
9	1 2 7 8 9	1 2 7 8 9

Let's try focused summaries for *Maria*, and *the child*. *Maria* is referred in *edu*s 1, 7, 8, and 9 (see example 2 and figure 4). The longest vein expression of these *edu*s (1 2 7 8 9) is the same in both golden and computed tree. Therefore, the summary focused on Maria will be:

Maria went alone to the market because Simon had to stay at home with the baby. I think she has a lot of trust in him to let him alone with the child. You know how Maria is: she is not very hurried to give credit to anybody.

The child is referred in *edu*s 2 and 7. The longest vein expression of these units is 1 2 7 in the golden tree and 1 2 7 9, in the computed tree. The summary focused on *the child* will be:

Maria went alone to the market because Simon had to stay at home with the baby. I think she has a lot of trust in him to let him alone with the child. She is not very hurried to give credit to anybody.

7 Data and experiments

The assumption on the correlation of vein structure with co-references was based on earlier experiments reported in (Cristea *et al*. 1998). An average, the results of experiments on Romanian and English texts revealed that in 99.1% references obey this conjecture.

Around 50 manually discovered cue-phrase patterns were used in the sentence-level tree construction, described in section 5. In order to validate the approach we developed two experiments. In the first experiment, *sdt*s were built based on the information given by an FDG parser, and in the second, *sdts* were generated based on our approach. We used a two pages excerpt from the original English version of G. Orwell's "1984" which contained 45 sentences out of which 19 were one-clause sentences, our attention being focused on the remaining 26 complex sentences. In the first experiment we used the FDG output both for extracting the units (the clauses) and for building the tree. It turned out that only 7 sentences (27%) could be resolved correctly while another 6 were only partially correct. In the second experiment for 20 sentences (76%) the method correctly indicated a unique *sdt*, while for the remaining 6 sentences more than one tree could be generated.

To validate the focused summarisation method guided by veins, a one-page text from "The Legends of Mount Olympus" of Al. Mitru, consisting of 62 *edu*s was used. 57 students, participants of the EUROLAN'01 summer school, were asked to extract a summary of the text, focused on *Hefaistos*, a secondary character in the extract. We then built a golden summary composed of units voted by more than a half of judges - 28 out of 57. For each judge, the recall and precision values were calculated. In the following table, the average of these values and the VT results are presented:

	Judges' results	VT's results
Precision	74.26%	73.33%
Recall	72.92%	64.71%

8 Discussions and further work

We are aware that errors can intervene in all processing steps of the described summarisation method (segmentation in *edu*s, detection of *sdt*s, anaphoric links detection). Further investigation will have to identify the overall trust in the method proposed.

An earlier investigation (Ide and Cristea, 2000) showed a correlation between the type of the anaphor (pronominal, proper nouns, definite or indefinite noun) and the percentage on which the antecedent is found along veins of the discourse structure. This suggests that the method of building the global structure of the discourse guided by references could be further sophisticated by using scores to account for type of antecedents.

The described method of inferring the discourse structure is deterministic in the sense that only one tree is obtained. Further development would have

to transform it into a beam-search type of processing, close to the one described in (Cristea, 2000), in order to combine contribution from cue-phrases, and references with that given by centering. This way, the problem itself of partial trees proliferation caused by cue-phrases with multiple patterns, presently ignored, could also be tackled.

As demonstrated by the example in the previous section, the computed vein expressions have a tendency to be larger than needed, this yielding to longer summaries. More sophisticated integration rules, automatically discovered from a discourse structure annotated corpus by learning, could fix this problem.

Finally, it is to note that the structure of a discourse as a complete tree gives more information than properly needed (at least for summarization purposes). An underspecified type of representation, keeping, for instance, only vein expressions not the whole tree, could be a better solution.

References

Ait-Mohtar, S. and Chanod, J.-P. 1997. Incremental Finite-State Parsing. *Proceedings of ANLP'97*, Washington.

Cole, R.A., Mariani, J., Uszkoreit, H, Zaenen, A. and Zue, V. 1995. Survey of the State of the Art in Human Language Technology.

Cristea, D. and Webber B.L. (1997). Expectations in Incremental Discourse Processing. *Proceedings ofACL/EACL'97*, Madrid.

Cristea, D., Ide, N. and Romary, L. (1998). Veins Theory: A Model of Global Discourse Cohesion and Coherence, *Proceedings of Coling/ACL'98*, Montreal.

Cristea, D., Ide, N., Marcu, D. and Tablan, V. 2000. Discourse Structure and Co-Reference: An Empirical Study. *Proceedings of The 18th International Conference on Computational Linguistics COLING'2000*, Luxembourg.

Cristea, D. and Dima, G.-E. 2001. An Integrating Framework for Anaphora Resolution. *Information Science and Technology*, Romanian Academy Publishing House, Bucharest, 4(3).

Cristea, D., Postolache, O.-D., Dima, G.-E. and Barbu, C. 2002a. AR-Engine – a framework for unrestricted co-reference resolution. *Proceedings of Language Resources and Evaluation Conference - LREC 2002*, Las Palmas, vol. VI: 2000-2007.

Cristea, D., Dima, G.E., Postolache, O.-D. and Mitkov, R. 2002b. Handling complex anaphora resolution cases, *Proceedings of the Discourse Anaphora and Anaphor Resolution Colloquium*, Lisbon.

Cristea,D. (2000): An Incremental Discourse Parser Architecture, D. Christodoulakis (Ed.) *Proceedings of the Second International Conference - Natural Language Processing - NLP 2000*, Patras, Greece, June 2000. Lecture Notes in Artificial Intelligence 1835, Springer.

Fox, B. 1987. Discourse Structure and Anaphora, Cambridge University Press.

Grosz, Barbara J., Aravind K. Joshi, Scott Weinstein. 1995. Centering: a Framework for Modelling the Local Coherence of Discourse. Computational Linguistics, 21(2).

Ide,N., Cristea,D. (2000): A Hierarchical Account of Referential Accessibility. *Proceedings of The 38th Annual Meeting of the Association for Computational Linguistics, ACL'2000*, Hong Kong.

Knott, A. and Dale, R. 1992. Using Linguistic Phenomena to Motivate a Set of Coherence Relations. Discourse Processes 18(1).

Mani, I. 2001. Automatic Summarization. Natural Language Processing series. John Benjamins Publishing Co., Amsterdam.

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

Puscasu, G. forthcoming. Elementary discourse unit segmentation. Dissertation thesis. "Al.I.Cuza" University of Iasi.

Soricutu and Marcu (2003) Sentence Level Discourse Paring using Syntactic and Lexical Information, *Proceedings of HLT/NAACL – 2003*, Edmonton.

Tufiş, Dan 1999. Tiered Tagging and Combined Classifiers. F. Jelinek, E. Nöth (Eds) *Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence, 1692*, Springer.

Vonk, W., Hustinx, L. and Simons, W. 1992. The Use of Referential Expressions in Structuring Discourse. *Language and Cognitive Processing*. 7 (3-4).