Tạp chí Tin học và Điều khiển học, T.23, S.3 (2007), 212–230

AN APPROACH IN AUTOMATICALLY GENERATING DISCOURSE STRUCTURE OF TEXT

LE THANH HUONG

Faculty of Information Technology, Hanoi University of Technology

Abstract. This paper presents a system for automatically generating the discourse structure of text. The system is divided into two levels: sentence-level and text-level. At the sentence-level, the discourse analyser uses sentential syntactic structures and cue phrases to derive discourse structures of sentences. A syntactic parser was integrated into the system to get the syntactic structure of sentences. This approach prevents combinatorial explosions while still generating accurate analyses. At the text-level, constraints about textual adjacency and textual organisation are integrated in a beam search to reduce the search space of the discourse analyser and to generate the best discourse structure. To signal discourse relations, beside the recognition factors that have been used by other research (e.g., overt cue phrase, cohesive devices), we propose two new factors – noun-phrase cues and verb-phrase cues. Our experiments with documents from the RST Discourse Treebank received 89.4% F-score for the discourse segmentation, 52.4% F-score for the sentence-level discourse analyser, and 38.1% F-score for the final output of the system. This approach provides good performance compared to existing discourse analysing systems.

Tóm tắt. Bài báo này giới thiệu một hệ thống tự động phân tích cấu trúc diễn ngôn của văn bản. Hệ thống được chia làm hai mức: mức câu và mức văn bản. Tại mức câu, hệ thống sinh cấu trúc diễn ngôn của câu dựa trên cấu trúc cú pháp của câu và các từ nối. Một bộ phân tích cú pháp được tích hợp vào hệ thống để sinh cấu trúc cú pháp cho câu. Cách tiếp cận này tránh được sự bùng nổ tổ hợp trong khi vẫn đưa ra được phân tích chính xác. Tại mức văn bản, các ràng buộc về sự liền kề và cấu trúc văn bản được kết hợp với phép tìm kiếm kiểu chùm (beam search) để giảm không gian tìm kiếm trong việc sinh cấu trúc diễn ngôn. Trong việc nhận dạng quan hệ diễn ngôn, bên cạnh các nhân tố nhận dạng đã được các nhà nghiên cứu sử dụng (như từ nối, các yếu tố liên kết), chúng tôi đề xuất hai yếu tố mới: các từ tín hiệu trong danh ngữ và động ngữ. Khi thử nghiệm với các văn bản trong tập ngữ liệu phân tích cấu trúc diễn ngôn RST-DT, hệ thống đạt được độ chính xác 89.4% F-score cho mức xác định các đơn vị diễn ngôn, 52.4% F-score cho mức phân tích cấu trúc diễn ngôn của câu và 38.1% F-score với mức cấu trúc diễn ngôn của toàn văn bản. Cách tiếp cận này đem lại kết quả tốt so với các hệ thông phân tích cấu trúc diễn ngôn hiện có.

Key words: Rhetorical Structure Theory, discourse analyser, syntactic structure, cue phrase, cohesive device, beam search.

1. INTRODUCTION

Many recent studies in Natural Language Processing have paid attention to a method of structured description of text called Rhetorical Structure Theory (RST). This theory was proposed by Mann and Thompson [11] and was developed by other researchers such as Hovy

[8], Marcu [12], and Forbes et al. [5]. Rhetorical structures, also called discourse structures, have been found to be useful in many fields of text processing such as text summarisation [12, 18], text translation [13], and text understanding [20,24]. However, only a few algorithms for implementing discourse analysers have been proposed so far. The amount of research available in discourse segmentation is considered small. In discourse analysing it is even smaller with most research in this field concentrating on specific discourse phenomena [7, 21].

A pioneering work in discourse analysing was proposed by Marcu [12]. His discourse analyser uses cue phrases as a signal to segment text and to recognise discourse relations, but this faces problems when cue phrases are not present in the text. Marcus system also produces an enormous number of redundant trees during its process. As the number of relations increases, the number of possible discourse trees increases exponentially.

Soricut and Marcu [23] introduced a sentence-level discourse parser called SPADE. It includes two probabilistic models: a discourse segmenter that identifies elementary discourse units and a sentence-level discourse parser that builds sentence-level discourse trees. Lexical and syntactic features are used in these models. The discourse segmenter consists of a statistical model, which assigns a probability to the insertion of a discourse boundary after each word in a sentence, and a segmenter, which uses the probabilities computed by the statistical model for inserting discourse boundaries. The input to the sentence-level discourse parser is a lexicalized syntactic parse tree in which the discourse boundaries have been identified. This parser assigns a probability to every potential candidate parse tree and then finds the best discourse tree. SPADE provides the best performance among existing sentence-level discourse analysers that we know of.

Our research aims at implementing a discourse analyser that automatically generates discourse structures of text. The system is called a Discourse Analysing System (DAS). We focus on improving the correctness of discourse segment boundaries; exploring new factors to recognise discourse relations; reducing the combinatorial explosion in searching for the best discourse structure; and improving the efficiency of the discourse analyser.

The rest of this paper is organised as follows. An overview of the rhetorical structure theory is given in Section 2. The discourse segmentation process of DAS is described in Section 3. Our method of recognising discourse relations is discussed in Section 4. The approach of constructing discourse trees is presented in Section 5. In Section 6, experiments are described and the results we have achieved so far are discussed. Section 7 concludes the paper and proposes possible future work.

2. RHETORICAL STRUCTURE THEORY - AN OVERVIEW

Rhetorical Structure Theory is a method of representing the coherence of text. It models the rhetorical structure of a text by a hierarchical tree that labels rhetorical relations between spans. This hierarchical tree diagram is called a "*rhetorical tree*", "*discourse tree*", or "*RST tree*". A leaf of an RST tree corresponds to an elementary discourse unit (EDU), which are clauses or clause-like units with independent functional integrity, whereas the internal tree nodes correspond to larger spans.

Figure 1 represents the discourse tree of Example (1). Instead of displaying the full text of each internal tree node, we cite the first and last EDUs that contribute to it (e.g., "1-2", "1-3"). An internal tree node contains one or several names (e.g., *Circumstance, Explanation*)

of the discourse relations that hold between adjacent, non-overlapping spans. The span that participates in a discourse relation is either a *nucleus* (N) or a *satellite* (S). The nucleus plays a more important role than the satellite in respect to the writers intention. If both spans have equal roles, they are both considered as nuclei in the relation.

(1) You should meet Peter today after you finish this work. He will go to Edinburgh tomorrow.

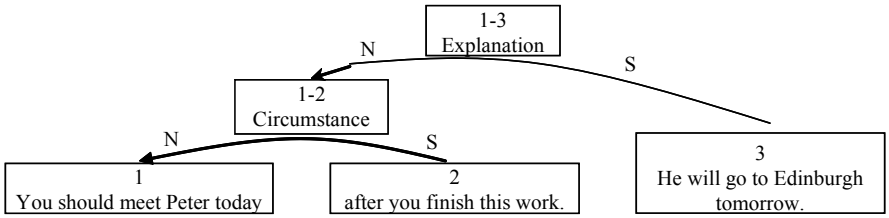

Figure 1. The Discourse Tree of Example (1)

To construct the rhetorical structure of a text, the following tasks should be performed: (i) segmenting text into EDUs; (ii) recognising discourse relations between spans; and (iii) selecting and combining discourse relations created in step (ii) to form a rhetorical structure that covers the entire text. The first task of our discourse analysing system discourse segmentation is presented next.

3. DISCOURSE SEGMENTATION

The purpose of discourse segmentation is to split a sentence into EDUs. Some research has been based on cue phrases to identify EDUs (e.g., [12]). However, Redeker [17] found that only 50% of clauses contain cue phrases. Therefore, segmentation based on cue phrases alone is not sufficient. Since an EDU is a clause or clause-like unit with independent functional integrity, syntactic information is useful for the segmentation process. In this research, both syntactic structures and cue phrases are used to solve this task. The usages of these factors are introduced in Sections 3.1 and 3.2 below.

3.1. Discourse Segmentation by Syntax - Step 1

This process splits a sentence into discourse segments using the syntactic structure of the sentence; documents from the Penn Treebank [15] were used to get the syntactic structure of sentences. Based on the syntactic structures, the discourse segmenter checks segmentation rules to split a sentence into discourse segments. The segmentation rules in this step are based on the segmentation principles proposed by Carlson et al. [1]. This paper proposes a method that automatically detects discourse segments, instead of a segmentation process that depends on humans as in [1]. In this paper, we analyse one of the segmentation principles to illustrate this process.

Principle (i) - *The clause that is attached to a noun phrase (NP) can be recognised as an embedded unit.*

The syntactic chains that correspond to principle (i) are:

(i-a) (NP ⟨text1⟩ (X ⟨textx⟩)* (SBAR|RRC ⟨text2⟩))

(i-b) (NP ⟨text1⟩ (X ⟨textx⟩)* (PRN ⟨text2⟩ (Y ⟨texty⟩)* (S ⟨text3⟩)))

(i-c) (NP ⟨text1⟩ (X ⟨textx⟩)* (PP ⟨text2⟩ (Y ⟨texty⟩)* (S|VP ⟨text3⟩)))

NP, SBAR, RRC, PRN, S, PP, VP stand for noun phrase, subordinate clause, reduced relative clause, parenthetical, sentence, prepositional phrase, and verb phrase respectively. '|' stands for 'or'. ⟨text1⟩, ⟨text2⟩, and ⟨text3⟩ stand for text. (X ⟨textx⟩)* and (Y ⟨texty⟩)* stand for any syntactic string (or none of them).

DAS finds the segmentation principle that maps to the syntactic structure of the input sentence and generates segment boundaries at the beginning of the NP, at the beginning of the SBAR—RRC—PRN—PP, and at the end of the NP. The correctness of segment boundaries is then checked by a post processing procedure.

Let us analyse Example (2):

(2) [Mr. Silas Cathcart built a shopping mall on some land][he owns]

DAS finds the segmentation rule that maps to the syntactic structure of the input sentence and generates segment boundaries. The correctness of segment boundaries is then checked by a post processing procedure. In Example (2), DAS derives an embedded unit "he owns" from the noun phrase "some land he owns". Based on the sentential syntactic information, the post processing procedure detects that "Mr. Silas Cathcart built a shopping mall on" is not a clause without the noun phrase "some land". Therefore, these two spans are combined into one. The sentence is now split into two discourse segments "Mr. Silas Cathcart built a shopping mall on some land" and "he owns". Besides splitting sentences into discourse segments, the segmenter also provides initial information about discourse relations. A discourse relation is initiated between the two segments in Example (2). The relation name and the span nuclearity are determined later in a relation recognition process (Section 4).

3.2. Discourse Segmentation by Cue Phrases - Step 2

Several noun phrases are considered as EDUs when they are accompanied by a strong cue phrase (Examples 3). These cases cannot be detected by syntactic information. Therefore, another segmentation process is integrated into DAS to deal with such cases. This process searches for a strong cue phrase in each discourse segment generated by Step 1. When a strong cue phrase is found, the algorithm splits the discourse segment into two EDUs: one unit is the noun phrase that contains the strong cue phrase, and another unit is the rest of the discourse segment. The set of strong cue phrases used in the experiments described in this research are: *according to, as a result of, although, because of, but also, despite, despite of, in spite of, irrespective, not only, regardless, without, –*. It is created basing on previous research about elementary discourse units such as [12] and on the observation of the annotated documents from the RST Discourse Treebank [19]. There are two cases that are treated differently by DAS, as presented by Examples (3) and (4):

(3) [*According to* a Kidder World story about Mr. Megargel,][all the firm has to do is "position ourselves more in the deal flow"].

(4) [In 1988, Kidder eked out a 46 million profit,][mainly because of severe cost cutting].

In the first case, there is no adverb that is left adjacent to the strong cue phrase (Example 3). A new EDU is created from the beginning position of the cue phrase to the end boundary of the noun phrase. The end boundary of a noun phrase is identified by a punctuation such as a comma, a semicolon, or a full stop. In the second case, some adverbs are left-adjacent to the strong cue phrase (Example 4). If these adverbs do not belong to the syntactic structure of the left part of the old discourse segment, a new EDU is created from the left most position of these adverbs to the end boundary of the noun phrase. Otherwise, the new EDU is created

in the same way as in the first case.

Similar to Step 1, Step 2 also initiates discourse relations between EDUs that it derives. The relation name and the span nuclearity are posited later in a relation recognition process, which is discussed next.

4. RECOGNISING DISCOURSE RELATIONS BETWEEN ELEMENTARY DISCOURSE UNITS

Although much effort has been put into empirical studies of recognising discourse relations [6,9,21], only a few algorithms for automatically positing discourse relations have been proposed so far. In this section, we introduce our algorithm and factors that are used by us in recognising relations.

4.1. Factors Used in Recognising Relations between EDUs

Research in relation recognition concentrates on several aspects of text cohesion such as cue phrases [4, 12, 21], anaphoric references [2, 16], and VP-ellipsis [9]. In this research, we applied several recognition factors that have been used by other researchers (overt cue phrases, reiterative devices, etc.) and proposed new recognition factors – noun-phrase cues and verb-phrase cues.

4.1.1. Overt Cue Phrases

Overt cue phrases (e.g., however, as a result), also called cue phrases, discourse connectives, conjunctions, or discourse markers, are words or phrases that connect text spans. Cue phrases have been the centre of research on discourse analysis due to two reasons. First, research on discourse has proved that cue phrases are used by the writer to construct the coherence of text [6,21]. Therefore, using cue phrases is an explicit way to express discourse structures. Second, identifying cue phrases is simple because it is essentially based on pattern matching. The cue phrase "when" in Example (5) determines a Circumstance relation between two clauses "He was staying at home" and "the police arrived".

(5) [He was staying at home][when the police arrived].

4.1.2. Noun-Phrase Cues and Verb-Phrase Cues

The two new recognition factors proposed by us are noun-phrase cues (NP cues) and verb-phrase cues (VP cues). Examples of NP cues and VP cues are shown below:

(6) [New York style pizza meets Californian ingredients,][and the result is the pizza from this Church Street pizzeria.]

(7) [By the end of this year, 63-year-old Chairman Silas Cathcart retires to his Lake Forest, Ill., home.][And that means 42-year-old Michael Carpenter will for the first time take complete control of Kidder.]

The noun "result" indicates a Result relation in Example (6); whereas the verb "means" signals an Interpretation relation between two sentences in Example (7). The phrases in the main noun phrases (i.e., subject or object) of a sentence that signal rhetorical relations are called NP cues. These phrases can be nouns, adjectives, adverbs, or their combination. For example, the adjective "following" in the noun phrase "the following week" may signal a Sequence relation. This word is considered as a NP cue. Similarly, the words in the main verb phrase of a sentence that signal relations are called VP cues.

NP cues, VP cues, and cue phrases are considered as separate recognition factors because

of their different behaviours in recognising relations. The same word in a NP, a VP, and a clause may signal different relations or may not signal any relation at all. Let us illustrate this statement using examples with the word "means". When "means" acts as a verb, it often signals an Interpretation relation (Example 7). When the noun "means" is in the main noun phrase of a sentence, it does not signal any relation (Example 8). Meanwhile, when the noun "means" is not in a main noun phrase of a sentence, but it is in the cue phrase "by means of", it indicates a Means relation (Example 9).

(8) [These means of transport are sometimes called accidental,][but this is not strictly correct.]

(9) [It is the magicians wand,][by means of which he may summon into life whatever form and mould he pleases.]

Overt cue phrases are identified based on pattern matching, whereas noun phrases or verb phrases have to be stemmed before being compared with the NP or VP cues. The sets of NP cues and VP cues were created by us, based on our research on different linguistic resources and on the RST Discourse Treebank [19].

4.1.3. Syntactic Information

According to Mann and Thompson [11], clausal relations reflect rhetorical relations within a sentence. For example, the rhetorical relation between a main clause and its subordinate clause is an asymmetric relation, in which the main clause is the nucleus, and the subordinate clause is the satellite. This idea is applied in DAS to posit the span nuclearity and to eliminate unsuitable relations. If two clauses are coordinate, their relation can be symmetric or asymmetric. Syntactic information can also be used to signal relation names. For example, the reporting and reported clauses of a sentence are considered as the satellite and the nucleus in an Elaboration relation:

(10) [Mr. Carpenter says][that Kidder will finally tap the resources of GE].

In Example (10), the reporting clause "Mr. Carpenter says" is considered as the satellite, whereas the reported clause "that Kidder will finally tap the resources of GE" is considered as the nucleus.

4.1.4. Time References

Discourse connection can be established by time relations between spans. If the time of a narrative changes from the present to the past, it is likely that the writer refers to a previous event that is the cause, the hypothetical, or the elaboration of the current event. A Cause relation holds between two sentences in Example (11).

(11) Mark has a terrible headache today. He drank too much last night.

If the time of the second span covers the time of the first span, a Circumstance relation usually holds (Example 12).

(12) Mark knows every person in this village. He has been living here for more than ten years.

The time reference can also be used to check the validity of a relation (see Section 4.3.2). Since the time reference can signal discourse connection and limit possible relations, it is combined with other factors to posit rhetorical relations, as described in Section 4.3.

4.1.5. Reiterative devices

The reiterative devices investigated in this research include word repetition, synonyms (employer/boss), hypernyms/hyponyms (country/Mexico), co-hyponyms (United Kingdom/Mexico),

and antonyms (simple/complex). Word repetition and synonyms are used to detect the discourse connections and discourse relations. For example, a Contrast relation often occurs when most words in two spans are similar and one span contains the word "not". A multinuclear relation (Contrast, List) often exists between spans whose main noun phrases are co-hyponyms or antonyms.

In order to recognise the reiteration devices, the main noun phrases (i.e., subjects and objects of sentences), verb phrases, and adjective phrases are extracted from the syntactic structure of spans. These phrases are then stemmed (e.g., "books" is converted into "book"). Next, DAS computes the semantic relation between these phrases using a thesaurus (WordNet [25]). The relations needed to be computed are word repetition, synonyms of nouns, hypernyms of nouns, co-hyponyms and antonyms of nouns, verbs, adjectives, and adverbs.

4.2. The Set of Relations

To generate discourse structures from texts, it is important to define a set of relations that is used to posit discourse relations. According to Mann and Thompson [11], the set of discourse relations is an open set. It can be modified for the purposes of particular genres and cultural styles. If the relation set consists of just a few relations, the discourse trees will be easier to construct, but they will not be informative. On the other hand, if it is a large set, the trees will be informative, but they will be difficult to build. The number of relations proposed by researchers varies from two [6] to over a hundred [1]. A problem arising here is how to justify whether one relation set is adequate or not, and how to justify whether one set is more appropriate than another. According to Knott [10], in order to justify a relation set, we have to have a way of deciding on an appropriate level of detail. Mann and Thompson [11] use five different relations to describe causal relations (Volitional Cause, Non-Volitional Cause, Volitional Result, Non-Volitional Result, and Purpose). All these five relations are grouped by Scott and de Souza [22] for the task of textual realisation.

The articles from the RST Discourse Treebank [19] used in this research were manually analysed using 110 different relations. It is very difficult to automatically construct RST trees based on such a large set. Therefore, we propose a smaller set by merging relations with similar characteristics from these 110 relations, resulting in a set of 22 relations: List, Sequence, Condition, Otherwise, Hypothetical, Antithesis, Contrast, Concession, Cause, Result, Cause-Result, Purpose, Solutionhood, Circumstance, Manner, Means, Interpretation, Evaluation, Summary, Elaboration, Explanation, and Joint. The difference among Cause, Result and Cause-Result is the span nuclearity in the relation. This set is created by considering the relations that are used most frequently in other research on discourse analysis (e.g., [2, 11, 12]).

We make no claim that our relation set covers all other relations or is correct in all details. It can be reduced, extended, or modified depending on different purposes and data. The modification of the relation set does not affect the approach used in this research. DAS is easily modified to fit with the new relation set by changing the conditions for recognising relations based on recognition factors proposed in Section 4.1. Other analysing modules used in DAS, i.e., discourse segmentation (Section 3) and discourse structure generation (Section 5), still remain the same since they are independent of the relation set.

4.3. Relation Recognition

DAS uses two types of conditions to recognise discourse relations. The conditions that are used to signal relations are called heuristic rules. The conditions that are used to check the

validity of a relation are called necessary conditions. The heuristic rules are the applications of recognition factors to a specific relation. For example, the heuristic rule that is used to recognise a List relation "The right span contains List cue phrases" (Section 4.3.2) is the application of the recognition factor cue phrases. The purpose of separating two kinds of recognition conditions is to reduce the work-load of the recognition process. To posit relations, DAS starts by finding recognition factors from spans. If these factors are strong enough to signal a relation (i.e., the total scores of the heuristic rules contributing to that relation are more than or equal to a threshold ((see Section 4.3.1)), then the necessary conditions of that relation will be checked. That relation will be posited if all necessary conditions are satisfied (see Section 4.3.2). Since a factor often signals a limited number of relations, DAS does not need to check all relations from the relation set.

A description of the process to recognise the List relation presented in Section 4.3.2 will further illustrate this idea. Before that, let us introduce our method of scoring heuristic rules and computing the score of a relation based on all evidence that contributes to the recognition of that relation.

4.3.1. Scoring Heuristic Rules

Cue phrases, NP cues, VP cues, and cohesive devices have different strengths in recognising rhetorical relations. The cue phrases explicitly signal discourse relations most of the time. Meanwhile, time reference mainly can signal discourse connection, it rarely can determine discourse relation name. Therefore, the heuristic rules using cue phrases are stronger than the heuristic rules using reiterative devices. To control the influence of these factors to the relation recognition, each heuristic rule is assigned a heuristic score. The rules involving cue phrases have the highest score of 100 because a cue phrase is the strongest factor to signal relations. NP cues and VP cues are also strong factors but weaker than cue phrases since they do not express relations in a straightforward way like cue phrases. As a result, the heuristic rules involving NP cues and VP cues are assigned a score of 90. The heuristic rules corresponding to the remaining recognition factors receive scores ranging from 20 to 80 since these factors are weaker than NP cues and VP cues.

In this research, we separate two types of scores: the score of a heuristic rule and the score of a specific cue phrase, NP cue, and VP cue. The heuristic rule involving cue phrases has the score of 100, which means DAS is one hundred per cent certain that the relation signalled by the cue phrase holds. However, it is only correct when that cue phrase explicitly expresses a relation.

Each cue phrase has a different level of certainty in signalling relations. In DAS, each cue phrase is assigned with a score. The cue phrase "instead of" always signals a Contrast relation; it has a score of 1. "And" can be a cue phrase for a List or an Elaboration relation, its score for each of these relations are lower than 1.

That means the cue phrase rule that applies to the cue phrase "and" is not one hundred per cent certain that a List relation holds. In other words, the score of a cue phrase rule should be reduced when this rule is applied to a weak cue phrase. Since the score of a cue phrase is between 0 and 1, DAS computes the actual score of a heuristic rule involving cue phrases as follow:

Actual-score(heuristic rule) = Score(heuristic rule) * Score(cue phrase).

This treatment is also applied to NP and VP cues. The actual score of a heuristic rule involving a NP or VP cue is: Actual-score(heuristic rule) = Score(heuristic rule) * Score(NP

cue or VP cue).

The actual score of other heuristic rules that do not involve cue phrase, NP or VP cue is:

Actual-score(heuristic rule) = Score(heuristic rule)

If several heuristic rules of a relation are satisfied, the score of that relation will be the total scores of all factors contributing to that relation.

Total-heuristic-score = Σ Actual-score (heuristic rule).

At present, heuristic scores are assigned by human linguistic intuitions. They can be optimised by an automatic training process. Unfortunately, we know of no discourse corpus that is large enough for this training purpose. Therefore, this training process has not been addressed in this research.

DAS seeks the recognition factors in the following order: cue phrases, NP cues, VP cues, and the remaining recognition factors. A rhetorical relation will be posited if the total-heuristic-score of this relation is greater than or equal to a threshold (. Choosing a reasonable value for this threshold is very important since a modification of this value may affect decisions in positing relations, therefore changing the rhetorical structure of a text. The threshold is assigned the score of 30 (compare to 100 as the maximum score of a heuristic rule), as by experiments we found that recognition factors can be very weak in many cases. For a better use of the threshold, a training method to optimise this value will be considered in future work. A sample of the recognition process representing the criteria to recognise a List relation is introduced next.

4.3.2. List Relation

A List relation is a multi-nuclear relation whose elements can be listed. The necessary conditions for a List relation between two discourse units, Unit1 and Unit2 (Unit1 precedes Unit2) are shown in Table 1. The first condition in Table 1 checks the linkage between these units by using reiterative and co-reference devices. Syntactic and semantic information determine the subject of these units and their relations. The second condition distinguishes a List relation from a Sequence relation.

Table 1. Necessary conditions for the List relation

Index	Necessary Condition
1	If both units have subjects and do not contain attribution verbs, then these subjects need to meet the following requirement: they must either be the same, identical, synonyms, co-hyponyms, hypernym/hyponym, or the subject of Unit2 is a pronoun or a noun phrase that can replace the subject of Unit1.
2	There is no explicit indication that the event expressed by Unit1 temporally precedes the event expressed by Unit2.

The heuristic rules for the List relation is shown in Table 2. We apply the criteria to recognise the List relation to Example (13).

(13) [Mr. Cathcart is credited with bringing some basic budgeting to traditionally free-wheeling Kidder.$_{13.1}$] [He also improved the firms compliance procedures for trading.$_{13.2}$].

In Example (13), the cue phrase "also" signals a List relation between the sentences (13.1) and (13.2). Since only the heuristic rule 1 (Table 2) is satisfied, the total-heuristic-score is:

Total-heuristic-score = Actual-score(heuristic rule 1) = score(heuristic rule 1) * score("also").

Table 2. Heuristic rules for the List relation

Index	Heuristic rule	Score
1	Unit2 contains List cue phrases	100
2	Both units contain enumeration conjunctions (first, second, third, etc)	100
3	Both subjects of Unit1 and Unit2 contain NP cues	90
4	If both units contain attribution verbs, the subjects of their reported clauses are similar, synonyms, co-hyponyms, or hypernyms/hyponyms	80
5	If the subjects of two units are co-hyponyms, then the verb phrase of Unit2 must be the same as the verb phrase of Unit1, or Unit2 has the structure "so + auxiliary + sbj"	80

The cue phrase "also" has the score of 1 for the List relation, so the total-heuristic-score is $100 * 1 = 100 > \theta$. Therefore, the necessary conditions of the List relation are checked. The subject of the sentence (13.2), "he", is a pronoun, which replaces the subject of the sentence (13.1), "Mr. Cathcart" (condition 1). There is no evidence of an increasingly temporal sequence (condition 2). Therefore, a List relation is posited between the sentences (13.1) and (13.2).

5. CONSTRUCTING DISCOURSE TREES

Constructing discourse trees of a text can be considered as the problem of searching for the combination of rhetorical relations that best describes the text, given all possible relations that hold between spans. Section 5.1 presents our method of positing relations between large spans. In order to take advantages of the clausal relations within a sentence, we divide the task of constructing discourse trees of a text into two levels: sentence-level (Section 5.2) and text-level (Section 5.3), each of which is processed in a different way.

5.1. Positing Relations between Large Spans

An important task in constructing discourse trees is to posit relations between large spans, which often contain more than one edu. For example, DAS has to find rhetorical relations between two sentences in Example (1), "*You should meet Peter today after you finish this work*" and "*He will go to Edinburgh tomorrow*". We rewrite Example (1) here as Example (14) for reading convenience.

(14) [You should meet Peter today$_{14.1}$][after you finish this work.$_{14.2}$][He will go to Edinburgh tomorrow.$_{14.3}$]

Marcu [12] explains the relations that are held between large spans in terms of the relations that are held between edus. According to the strong compositionality criterion of Marcu, "*if a rhetorical relation R holds between two textual spans of the tree structure of a text, then it can be explained by a similar relation R that holds between at least two of the most important textual units of the constituent spans.*" From this point of view, Marcu analyses relations between large spans by considering only relations between their nuclei.

The edus (14.1) and (14.3) are the most important units of the first and the second sentences in Example (14) respectively. Therefore, according to Marcu, the relation between these sentences is the relation between (14.1) and (14.3). Since the span (14.3) explains for the information in the span (14.1), an Explanation holds between them. The span (14.1) is the nucleus and the span (14.3) is the satellite. Consequently, an Explanation holds between

spans (14.1-14.2) and (14.3), in which the span (14.1-14.2) is the nucleus and the span (14.3) is the satellite.

The compositionality criterion of Marcu skips recognition factors from the satellites of the constituent spans, which can also be used to signal relations between large spans. Example (15) illustrates this situation. Figure 2 shows the discourse tree that connects two sentences in Example (15). The name of the rhetorical relation between these sentences has not been recognised.

(15) [With investment banking as Kidder's "lead business," where do Kidder's 42-branch brokerage network and its 1,400 brokers fit in?$_{15.1}$][To answer the brokerage question,$_{15.2}$] [Kidder, in typical fashion, completed a task-force study.$_{15.3}$]

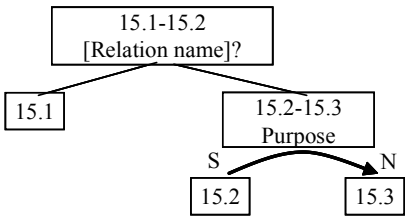

Figure 2. Discourse tree of Example (15)

The VP cue "To (+*verb*)" in span (15.2) indicates a Purpose relation between the two clauses (15.2) and (15.3), in which the span (15.2) is the satellite and the span (15.3) is the nucleus. The VP cue "*answer*" in the span (15.2) indicates a *Solutionhood* relation between two sentences; one is the span (15.1), another covers spans (15.2) and (15.3). If DAS applied the compositionality criterion of Marcu to Example (15), DAS would ignore the satellite (15.2). As a result, it would be difficult to recognise the relation that holds between these two sentences.

Example (15) shows that **although the content of a satellite does not determine rhetorical relations of its parent span, recognition factors that belong to the satellite are still valuable**. We noticed that cue phrases, NP cues, and VP cues of the left most EDU of both large spans can contribute to the relation between the large spans. Meanwhile, other cue phrases inside these spans contribute to the internal relations within each large span. For this reason, we propose a new recognizing criterion to detect discourse relations between large spans (rule 5). In recognising the relation between large spans, DAS does not use only the nuclei of the large spans as Marcu did, but also their first EDUs, whether they are nuclei or not.

We applied the compositionality criterion of Marcu and extended it for the case when a satellite stands at the beginning of the large span. To formalise the rules that are used to posit rhetorical relations between large spans, the following definitions are applied:

- $\langle$T$\rangle$ represents a span.

- $\langle T_i\ T_j\rangle$ represents a span that covers two adjacent, non-overlapping spans $\langle$Ti$\rangle$ and $\langle$Tj$\rangle$, which are related by a rhetorical relation. The possible roles of $\langle$Ti$\rangle$ and $\langle$Tj$\rangle$ in this relation are Nucleus Nucleus, Nucleus Satellite, or Satellite Nucleus. These states are encoded as $\langle T_i\ T_j \mid$ NN$\rangle$, $\langle T_i\ T_j \mid$ NS$\rangle$, and $\langle T_i\ T_j \mid$ SN$\rangle$, respectively.

rhet_rels($\langle\ T_i\ \rangle$, $\langle\ T_j\ \rangle$) represents the rhetorical relations between $\langle\ T_i\ \rangle$ and $\langle\ T_j\ \rangle$.

The rules used in DAS are presented in Table 3 below.

Table 3. DASs rule set using in recognising discourse relations

Index	Rule	Description
1	rhet_rels(<T_1 T_2 | NS>, <T>) ≡ rhet_rels(<T_1>, <T>)	If: there is a relation between <T_1> and <T_2>, in which <T_1> is the nucleus and <T_2> is the satellite; Then: rhetorical relations between span <T_1 T_2> and its right-adjacent span <T> are the relations that hold between <T_1> and <T>.
2	rhet_rels(<T>, <T_1 T_2 | NS) ≡ rhet_rels(<T>, <T_1>)	If: there is a relation between <T_1> and <T_2>, in which <T_1> is the nucleus and <T_2> is the satellite; Then: rhetorical relations between <T> and its right-adjacent span <T_1 T_2> are the relations that hold between <T> and <T_1>.
3	rhet_rels(<T_1 T_2 | NN>, <T>) ≡ rhet_rels(<T_1>, <T>) ∪ rhet_rels(<T_2>, <T>)	If: there is a relation between <T_1> and <T_2>; both <T_1> and <T_2> are nuclei Then: rhetorical relations between <T_1 T_2> and its right-adjacent span <T> are the relations that hold either between <T_1> and <T>, or between <T_2> and <T>.
4	rhet_rels(<T>, <T_1 T_2 | NN>) ≡ rhet_rels(<T>, <T_1>) ∪ rhet_rels(<T>, <T_2>)	If: there is a relation between <T_1> and <T_2> and both <T_1>, <T_2> are nuclei Then: rhetorical relations between <T_1 T_2> and its left-adjacent span <T> are the relations that hold either between <T> and <T_1>, or between <T> and <T_2>.
5	rhet_rels(<T>,<T_1 T_2| SN>) ≡ (rhet_rels signalled by unused cue phrases in <T_1>) ∪ rhet_rels(<T>,<T_2>)	If: there is a relation between <T_1> and <T_2>, in which <T_1> is the satellite and <T_2> is the nucleus Then: rhetorical relations between <T_1 T_2> and its left-adjacent span <T> are either the relations that hold between <T> and <T_2>, or the relations that are signalled by the unused cue phrases in <T_1>.

To posit rhet_rels(⟨T⟩, ⟨ T_1 T_2 | SN⟩), DAS first finds all cue phrases in span ⟨T1⟩ which have not been used to create the relation between ⟨T1⟩ and ⟨ T_2 ⟩, then checks rhet_rels(⟨T⟩, ⟨T1⟩) by using these cue phrases. If a relation is found, it is assigned to rhet_rels(⟨T⟩, ⟨ T_1 T_2 | SN⟩). Otherwise, rhet_rels(⟨T⟩, ⟨ T_1 T_2 | SN⟩) (rhet_rels(⟨T⟩ ⟨ T_2 ⟩).

Applying rule 5 to Example (15) with the spans (15.1) and (15.2-15.3), ⟨ T_1 ⟩ has one VP cue "*answer*" since the VP cue "*to*" is used to signal the relation between (15.2) and (15.3). The relation between (15.1) and (15.2-15.3) is recognised as *Solutionhood* by using the cue "*answer*" in restCPs. If DAS uses Marcus rules, rhet_rels((15.1), (15.2 15.3 | SN)) = rhet_rels((15.1), (15.3)). That means the VP cue "*answer*" is not considered in Marcus system.

5.2. Constructing Discourse Trees at the Sentence-level

This module takes the output of the discourse segmenter as the input and generates a discourse tree for each sentence. As mentioned in Section 3, the discourse segmenter has already generated EDUs and some information about rhetorical relations between EDUs. The sentence-level discourse analyser only has to posit relation names and the span nuclearity of discourse relations. It is achieved by using the rules described in Section 5.1 and the relation recognition method described in Section 4. Syntactic information and cue phrases are the main recognition factors for the recognition process at the sentence-level. For example, the rhetorical relation between a reporting clause and a reported clause in a sentence is an *Elaboration* relation. The reporting clause is the satellite; the reported clause is the nucleus of that relation (Example 16).

(16) [She said][she went to the British library yesterday].

Cue phrases are also used in DAS to signal discourse relations in a sentence, as shown in

Example (17):

(17) [He came late] [*because of* the traffic].

The cue phrase "*because of*" signals a relation between the clause containing this cue phrase and its left adjacent clause. The clause containing "*because of*" is the satellite of that relation. When syntactic information and cue phrases are not strong enough to signal discourse relations, the other recognition factors discussed in Section 4.1 are taken into account.

To construct the sentence-level discourse tree, after all relations within a sentence have been posited, all spans that correspond to a sub-tree are replaced by that sub-tree, such as in Example (18):

(18) [[She knows.$_{18.1}$] [what time you will come.$_{18.2}$]] [because I told her yesterday.$_{18.3}$].

The discourse segmenter outputs two sub-trees, one with two leaves "*She knows*" and "*what time you will come*"; another with two leaves "*She knows what time you will come*" and "*because I told her yesterday*". DAS combines these two sub-trees into one tree. With the presented method of constructing sentential discourse trees based on syntactic structures and cue phrases, combinatorial explosions can be prevented while DAS still gets accurate analyses.

5.3. Constructing Discourse Trees at the Text-level

The discourse trees at the text-level are generated by selecting and applying relations from all possible relations between large spans. Our method of reducing the search space for this problem is discussed in Section 5.3.1. The search algorithm of DAS is presented in Section 5.3.2.

5.3.1. Search Space

Previous research on discourse analysers shows that the search space of a normal discourse analyser is enormous [2, 12]. Therefore, a crucial problem in discourse analysing is to reduce the search space. We solved this problem by using constraints about textual organisation and textual adjacency, as discussed below.

Normally, each text has an organisational framework, which consists of sections, paragraphs, etc., to express a communicative goal. Each textual unit completes an argument or a topic that the writer intends to convey. Thus, a span should have semantic links to spans in the same textual unit before connecting with spans in a different one. Based on this idea, to generate the discourse tree of a text, instead of testing every possible combination of discourse trees, only discourse trees whose spans are in the same textual unit (a paragraph, a sub-section) are considered.

The second factor used in reducing the search space is the adjacent criterion of rhetorical structures. Since the spans that contribute to a rhetorical relation must be adjacent [11], only adjacent spans are considered to be connected in generating new relations. This search space is smaller than the search space reported in [12] since most discourse trees in his search space connect discourse trees that correspond to non-adjacent spans. Marcus system [12] generates all possible trees, and then uses the adjacent constraint to filter the inappropriate ones. We reduce the search space further by applying this constraint earlier, when the candidate solutions are generated, instead of filtering candidates after they are generated.

In DAS, relations between non-adjacent spans may be generated during the search process when they are parts of two larger discourse trees corresponding to adjacent spans. The discourse trees corresponding to the non-adjacent spans are not in the DAS search space. Instead, they are stored in the system in order to be called when they are needed. Figure 3 illustrates

a situation when the relation between two non-adjacent spans is called. $\langle T_1 \rangle$, $\langle T_2 \rangle$, $\langle T_3 \rangle$, $\langle T_4 \rangle$, $\langle T_5 \rangle$, $\langle T_6 \rangle$ are adjacent spans by this order.

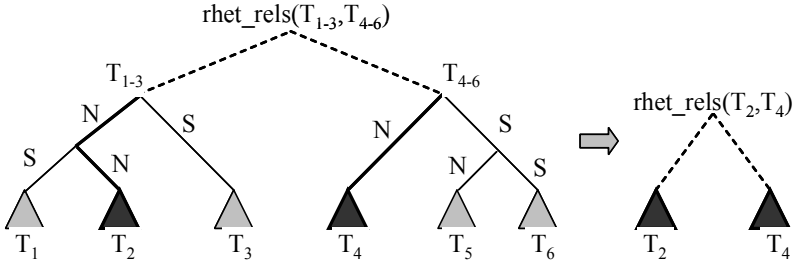

Figure 3. A situation when the rhetorical relation between two non-adjacent spans is called

In Figure 3, the relation between the two non-adjacent spans $\langle T_2 \rangle$ and $\langle T_4 \rangle$ is called when DAS attempts to find a relation between two adjacent spans $\langle T_i \rangle$ and $\langle T_j \rangle$. If the relation between $\langle T_2 \rangle$ and $\langle T_4 \rangle$ has not been generated before, it will be posited based on recognition factors mentioned in Section 4.

5.3.2. Search algorithm

To find the best combination of rhetorical relations, we apply a beam search, which minimises the search space while maximising the tree quality. A set called Subtrees is used to store sub-trees that have been created during the constructing process. The sub-trees in this set correspond to adjacent and non-overlapping spans. At the beginning, *Subtrees* consists of sentential discourse trees. As sub-trees corresponding to adjacent spans are connected to construct bigger trees, *Subtrees* contains fewer and fewer members. When *Subtrees* contains only one tree, this tree represents the rhetorical structure of the text.

All relations between adjacent spans that can be used to construct bigger trees at a step form a set *PotentialH*. Each relation of this set is assigned a score called the *total-heuristic-score*, which is equal to the total score of heuristic rules that signal the relation (see Section 4.3.1). To control the textual block level (paragraph, section, etc.), each relation is assigned a *block-level-score*, whose value depends on the block level of the spans that participate in the relation. The *block-level-score* and the *heuristic-score* are set in different value-scales so that the combination of sub-trees in the same textual block always has a higher score than that in a different textual block. If two sub-trees are in the same paragraph, the relation that connects these sub-trees will have the *block-level-score* = 0. (The paragraph is considered as the lowest block level.) If two sub-trees are in different paragraphs, and a value Li is the lowest block level where two sub-trees are in the same unit, the *block-level-score* of the relation corresponding to their parent tree is equal to -1000 * Li. For example, if two sub-trees are in the same section but in different paragraphs; and there is no subsection in this section; then Li is equal to 1. The negative value (-1000) indicates the higher the distance between two spans, the lower the combinatorial priority they get. The *block-level-score* of a relation is the lowest *block-level-score* among all relations between a sub-tree of the left node and a sub-tree of the right node. When selecting a relation, the relation with the higher *block-level-score* is preferred. If two or more relations have the same *block-level-score*, the one with the higher *total-heuristic-score* is chosen. A variable total-score is used to store the sum of the *total-heuristic-score* and the *block-level-score*.

An *accumulated-score* is used to store the value of the search path. The *accumulated-score* of a path at a step is the highest predicted-score of that path at the previous step.

A *predicted-score* of a relation at a step is equal to the sum of the *accumulated-score* of the previous step and the *total-score* of the relation. The search process now becomes the process of searching for the relation in *PotentialH* that has the highest *predicted-score*. If a relation involving two spans $\langle T_i, T_j \rangle$ is chosen, the new sub-tree created by joining the two sub-trees corresponding to spans $\langle T_i \rangle$ and $\langle T_j \rangle$ is added to *Subtrees*. The set *Subtrees* is now updated so that it does not contain overlapping discourse trees. The set *PotentialH* is also changed according to the change in *Subtrees*. The relations between the new sub-tree and its adjacent sub-trees in *Subtrees* are created and added to *PotentialH*.

All relations that have been computed are stored in the system to assure that a discourse tree will not be created twice. When detecting a new relation, the analyser first checks whether it has been created or not. If it is not, it will be posited based on the conversional rules (Section 5.1) and different relation recognition factors (Section 4.1). If no relation is recognised between two discourse sub-trees, a *Joint* relation is assigned. Thus, a discourse tree that covers the entire text can always be found.

The beam width M is chosen to be 10 in DAS since through experiments it was found to be large enough to derive good discourse trees. If at a later stage it was found that this value is insufficient, DAS only needs to increase this value. All other values are updated accordingly. If *Subtrees* contains only one tree, this tree is added to the tree set. This set is used to store the discourse trees that cover the entire text. The searching algorithm terminates when the number of discourse trees in the tree set is equal to the number of trees required by the user.

6. EVALUATION

To evaluate DAS, we carried out experiments using the documents from the RST Discourse Treebank [19] and computed the accuracies of the system on seven levels of processing, which is described in Section 6.1. Section 6.2 discusses the result we have achieved so far and compares DAS performance with the best performance among existing discourse systems.

6.1. Experimental description

The data used in our experiments were documents from the RST Discourse Treebank [19]. This corpus consists of 385 Wall Street Journal articles from the Penn Treebank [15]. These articles have been manually annotated with rhetorical structures in the RST framework using 110 different rhetorical relations. There are 53 articles in the corpus that have been independently annotated by a second analyst. These 53 documents were used in DAS to compute the human agreement on the rhetorical structures derived from the same texts.

Since 110 relation names are used to annotate discourse relations in RST corpus and 22 relations are used in DAS, mapping relation names between these two sets is necessary. Before comparing the discourse trees generated by DAS with the discourse trees in the RST corpus, each relation name from the RST corpus was converted into a correspondent relation name in DAS.

The syntactic information of the documents used in our experiments was taken from the Penn Treebank, which was used as the input to the discourse segmenter. The Penn Treebank was chosen because of two reasons. First, documents from the RST Discourse Treebank, which were used in experiments of this research, are also taken from the Penn Treebank. Second, this corpus is widely accepted and used in much syntactic research.

The accuracy of the output of DAS was measured at seven levels. The output of one

process was used as input to the process following it.

• Level 1 - The accuracy of discourse segments. This was calculated by comparing the segment boundaries assigned by the discourse segmenter with the segment boundaries assigned by a human.

• Level 2 - The accuracy of the combination of spans at the sentence-level. DAS generates a correct combination if it connects the same spans as the human does.

• Level 3 - The accuracy of the span nuclearity at the sentence-level.

• Level 4 - The accuracy of rhetorical relations at the sentence-level.

• Level 5 - The accuracy of the combination of spans for the entire text.

• Level 6 - The accuracy of the span nuclearity for the entire text.

• Level 7 - The accuracy of rhetorical relations for the entire text.

The systems performance is represented by precision, recall, and F-score. The precision is the proportion of assignments made that were correct. The recall is the proportion of possible assignments that were actually assigned. The F-score is a measure combining precision and recall into a single figure. We use the version in which they are weighted equally, defined as 2*precision*recall/(precision+recall).

The performance of DAS is shown in Table 4. The performance of the human was considered as the upper bound for DAS performance. This value was obtained by evaluating the agreement between human annotators using the 53 double-annotated documents from the RST corpus. The performance of humans is also shown in Table 4. An evaluation of these performances is presented in Section 6.2.

Table 4. DAS performance Vs. human performance

Level		1	2	3	4	5	6	7
DAS	Precision	90.7	69.7	61.9	53.4	54.9	46.8	38.6
	Recall	88.1	67.2	59.8	51.5	53.4	45.5	37.6
	F-score	**89.4**	**68.4**	**60.8**	**52.4**	**54.2**	**46.2**	**38.1**
Human	Precision	98.7	88.4	82.6	69.2	73.0	65.9	53.0
	Recall	98.8	88.1	82.3	68.9	72.4	65.3	52.5
	F-score	**98.7**	**88.3**	**82.4**	**69.0**	**72.7**	**65.6**	**52.7**
$\frac{F-score(DAS)}{F-score(Human)}$*100%		90.6	77.5	73.8	75.9	74.6	70.4	72.3

6.2. Discussion

In the experiments carried out in this research, the output of one process was used as input to the process following it. The error of one process is, therefore, the accumulation of the error of the process itself and the error from the previous process. As a result, the accuracy of DAS and that of humans decline as the processing level increases. DAS provides a reliable result at the discourse segmentation level (90.7% precision and 88.1% recall). The systems performance at the sentence-level is acceptable when compared with humans. The low accuracies of DAS for the entire text (46.2% F-score at Level 6 and 38.1% F-score at Level 7) indicate that the discourse trees generated by DAS are much different from those in the corpus. We found that some documents used in these experiments contain incorrect paragraph boundaries. This problem contributes to the error of DAS output at the text-level.

Table 4 shows that the accuracy of the discourse trees given by human agreement is not high

either (52.7% F-score). Perhaps it is because discourse relations are too complex. Different people may create different discourse trees for the same text [11]. Because of the multiplicity of RST analyses, the discourse analyser should be used as an assistant rather than a stand-alone system. For that purpose, DAS output is designed to be editable by normal users through a friendly human computer interface. The RST Tool created by O'Donnell [14] is used for this purpose.

To compare the system described in this paper with other research in discourse analysis, we compared DAS with the most recent high performance discourse systems SPADE. This is a sentence-level discourse analyser generated by Soricut and Marcu [23], which includes two probabilistic models that can be used to identify EDUs and build sentence-level discourse trees. The RST Discourse Treebank is also used in their experiment, in which 347 articles are used as the training set and 38 articles are used as the test set. The precision and recall of the SPADE segmenter when syntactic trees from the Penn Treebank are used as the input are 84.1% and 85.4%, respectively. The segmentation accuracy of SPADE is slightly lower than DAS. The difference between SPADE and DAS is that DAS combines syntactic information with cue phrases for discourse segmentation, instead of using lexical information in a probabilistic model as in SPADE. The above performances indicate that syntactic information is a good feature for discourse segmentation. At the sentence-level, the F-scores percentage of DAS performance and the performance of human analysts is approximate to that of SPADE.

To our knowledge, there is only one report about the accuracy of the discourse analyser at the text-level written by Marcu [12]. When using WSJ documents from the Penn Treebank, Marcus decision-tree-based discourse analyser received 21.6% recall and 54.0% precision for the span nuclearity; 13.0% recall and 34.3% precision for discourse relations. Therefore, DAS provides a better performance than the system created by Marcu [12].

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a discourse analysing system and evaluated it using the RST discourse corpus. The experiments show that syntactic information and cue phrases are efficient in constructing discourse structures at the sentence-level, especially in discourse segmentation (89.4% F-score). At the text-level, the constraints of textual adjacency and textual organisation are integrated in a beam search to reduce the search space. It is shown that the search space of DAS is much smaller than that of Marcu system [12]. The experiments show that the proposed approach can produce reasonable results compared to human annotator agreements.

To improve DAS performance, future work includes refining the segmentation rules; and investigating a method to identify the boundaries of high level textual units (paragraph, section, etc.). We propose to use an approach of topic segmentation (e.g., [3]) for the second problem mentioned above. A training method for optimising the value of the threshold (see Section 4.3.1) and different scores used in DAS (scores of heuristic rules, cue-phrases, NP cues, and VP cues) will be considered in future work. We would also like to integrate a syntactic parser to DAS and use the syntactic structures generated by this parser as the input to the discourse segmenter, instead of using the syntactic documents from the Penn Treebank [15] (see Section 3.1.1). We hope this research will aid in the future development of text processing such as text summarisation and text extraction.

Acknowledgment. The author gratefully acknowledges the receipt of a grant from the Flemish Interuniversity Council for University Development Cooperation (VLIR UOS) which enabled the research team to carry out this work.

REFERENCES

[1] L. Carlson, D. Marcu, and M. E. Okurowski, Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory, *Current Directions in Discourse and Dialogue*, Kluwer Academic Publishers, 2002.

[2] S.Corston-Oliver, "Computing Representations of the Structure of Written Discourse". PhD Thesis. 1998. University of California, Santa Barbara, CA, U.S.A.

[3] F. Choi, Advances in domain independent linear text segmentation, *Proc. of NAACL'00*, USA, 2000.

[4] K. Forbes and B. Webber, A semantic account of adverbials as discourse connectives, *Proc. of Third SIGDial Workshop*, Philadelphia PA, 2002.

[5] K. Forbes, E. Miltsakaki, R. Prasad, A. Sarkar, A. Joshi and B. Webber, D-LTAG System: Discourse Parsing with a Lexicalized Tree-Adjoining Grammar, *Journal of Logic, Language and Information* **12** (3) (2003) 261–279.

[6] B. J. Grosz and C. L. Sydner, Attention, intentions and the structure of discourse, *Computational Linguistics* **12** (1986) 175–204.

[7] J. Hirschberg and D. Litman, Empirical studies on the disambiguation of cue phrases, *Computational Linguistics* **19** (3) (1993) 501–530.

[8] E. Hovy, Automated Discourse Generation Using Discourse Structure Relations, *Artificial Intelligence*, Elsevier Science Publishers, Amsterdam, (63) (1993) 341–385.

[9] Kehler and S. Shieber, Anaphoric dependencies in ellipsis, *Computational Linguistic* **23** (3) (1997).

[10] Knott, "A Data-Driven Methodology for Motivating a Set of Coherence Relations." Ph.D. Thesis. 1996. University of Edinburgh, UK.

[11] W. Mann and S. Thompson, Rhetorical Structure Theory: Toward a Functional Theory of Text Organisation, *Text* **8** (3) (1988) 243–281.

[12] D. Marcu, The theory and practice of discourse parsing and summarization, *MIT Press*, 2000.

[13] D. Marcu, L. Carlson, and M. Watanabe, The Automatic Translation of Discourse Structures, *Proc. of NAACL'00, Seattle*, USA, 2000.

[14] M. O'Donnell, RSTTool, http://www.wagsoft.com/RSTTool/index.html, 2002.

[15] Penn Treebank, Linguistic Data Consortium, 1999, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC99T42

[16] M. Poesio and B. Di Eugenio, Discourse structure and anaphoric accessibility, *Proc. of the ESSLLI Workshop on Information Structure, Discourse Structure and Discourse Semantics*, Helsinki, 2001.

[17] G. Redeker, Ideational and pragmatic markers of discourse structure, *Journal of Pragmatics* (1990) 367–381.

[18] L. H. M. Rino and D. Scott, Automatic generation of draft summaries: Heuristics for content selection, *Proc. of CSNLP94*, Dublin, Ireland, 1994.

[19] RST-DT, RST Discourse Treebank, Linguistic Data Consortium, 2002, http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalog Id=LDC2002T07.

[20] L. Rutledge, B. Bailey, J. v. Ossenbruggen, L. Hardman and J. Geurts, Generating Presentation Constraints from Rhetorical Structure, *Proc. of the 11th ACM conference on Hypertext and Hypermedia*, San Antonio, Texas, USA, 2000 (19–28).

[21] D. Schiffrin, *Discourse markers*, Cambridge: Cambridge University Press, 1987.

[22] D. R. Scott and C. S. de Souza, Getting the message across in RST-based text generation, *Current Research in Natural Language Generation*, Academic Press, 1990 (47–73).

[23] R. Soricut and D. Marcu, Sentence level discourse parsing using syntactic and lexical information, *Proc. of HLT-NAACL,* 2003.

[24] M. Torrance and N. Bouayad-Agha, Rhetorical structure analysis as a method for understanding writing processes, *Proc. of the International Workshop on multi-disciplinary approaches of discourse (MAD 2001)*, 2001.

[25] M. Torrance and N. Bouayad-Agha, Rhetorical structure analysis as a method for understanding writing processes, *Proc. of the International Workshop on Multi-disciplinary Approaches of discourse (MAD 2001)*, 2001.

[26] M. Torrance and N. Bouayad-Agha, Rhetorical structure analysis as a method for understanding writing processes, *Proc. of the International Workshop on multi-disciplinary approaches of discourse (MAD 2001)*, 2001.

[27] WordNet, http://www.cogsci.princeton.edu/w̃n/index.shtml

Received on April 23 - 2007
Revised on August 24 - 2007