

AUTOMATED DISCOURSE SEGMENTATION BY SYNTACTIC INFORMATION AND CUE PHRASES

Huong Le Thanh, Geetha Abeysinghe, and Christian Huyck
School of Computing Science, Middlesex University,
The Burroughs, London NW4 4BT, UK.
{H.Le, G.Abeysinghe, C.Huyck}@mdx.ac.uk

Abstract

This paper presents an approach to automatic segmentation of text written in English into Elementary Discourse Units (EDUs)¹ using syntactic information and cue phrases. The system takes documents with syntactic information as the input and generates EDUs as well as their nucleus/satellite roles. The experiment shows that this approach gives promising results in comparison with some of the prominent research relevant to our approach.

Keywords: Natural Language Processing, Discourse Segmentation, Syntactic Information, Cue Phrases.

1 Introduction

Previous research in discourse has shown that the discourse structure of a text is constructed from smaller discourse segments ([1], [2]). According to Mann and Thompson [1], all discourse units should have independent functional integrity, such as independent clauses. The smallest discourse unit is called an Elementary Discourse Unit (EDU) [3].

Discourse has been automatically segmented using disparate phenomena: lexical cohesion ([4], [5], [6]), discourse cues ([2], [3], [7], [8]), and syntactic information ([9], [10]). However, the criteria to indicate the exact discourse segment boundaries are still not certain.

The weakness of the lexical cohesion approach is that it cannot guarantee independent discourse units, which is the essential condition for discourse segmentation. Discourse cues, such as cue phrases, pauses, and referential identities ([3], [11]) can be a solution for this problem. Marcu's shallow analyser [3] splits text into EDUs by mapping cue phrases and punctuation marks. However, this approach cannot correctly identify boundaries in complex sentences, which do not have any lexical discourse cues.

Passonneau and Litman [7] proposed two sets of algorithms for linear segmentation based on the linguistic features of discourse. The first set is based on referential pronoun phrases, cue words and pauses. The second set uses error analysis and machine learning. The machine learning method requires training, which is heavily dependent on the manually annotated corpora. A large discourse corpus for such a training purpose is difficult to find.²

One of the well-organised work, which used syntactic approach, was done by Corston-Oliver [10]. He defined a rule set for discourse segmentation basing on grammatical information. However, the computational algorithm used by him to segment text is not mentioned in his thesis. In addition, Corston-Oliver's system does not detect the cases when strong cue phrases make noun phrases become EDUs.

Considering the problems mentioned above, we propose a new method that combines the syntactic approach with the discourse cue approach. Since a typical discourse unit is an independent clause or a simple sentence [1], the text is first split into EDUs using syntactic information. To deal with the case where strong cue phrases make a noun phrase become a separate EDU, a further segmentation process is undertaken after segmenting by syntax. The purpose of this process is to detect strong cue phrases. These processes will be discussed in more detail in the following sections.

The rest of this paper is organised as follows. The first step of our system (Step 1), discourse segmentation by syntax, is described in Section 2. Discourse segmentation by cue phrases (Step 2) is represented in Section 3. In Section 4, we describe our experiment and discuss the result we have achieved so far. Section 5 concludes the paper and delineates the possible future work of this approach.

¹ For further information on "EDU", see [3].

² The biggest discourse corpus that we know of is the RST Discourse Treebank [12], with 385 Wall Street Journal articles.

2 Discourse Segmentation by Syntax – Step 1

The discourse segmentation by syntax module takes parsed documents from the Penn Treebank [13] as its input. One sentence is analyzed at each iteration of the segmentation process.³ This module not only splits sentences into clauses, but also provides primary information about discourse relations among EDUs, such as which EDUs should have a discourse connection, and the status assigned to them (nuclei and satellites).

2.1 Segmentation Principles

In Step 1, the principles for segmenting sentences into discourse units are based on the syntactic relations between words. These principles are based on previous research on discourse segmentation ([10], [14]). The main principles used in our system are shown below:

(i) *The clause that is attached to a noun phrase (NP) can be recognised as an embedded unit. If the clause is a subordinate clause, it must contain more than one word.*

For example:

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

(ii) *Coordinate clauses and coordinate sentences of a complex sentence are EDUs.*

For example:

(2) [The firm's brokerage force has been trimmed][and its mergers-and-acquisitions staff increased to a record 55 people.]

(iii) *Coordinate clauses and coordinate elliptical clauses of verb phrases (VPs) are EDUs. Coordinate VPs that share a direct object with the main VP are not considered as a separate discourse segment.*

For example:

(3) [The firm seemed to be on the verge of a meltdown,][racked by internal squabbles and defections.]

(iv) *Clausal complements of reported verbs and cognitive verbs are EDUs.*

For example:

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

Using the Penn Treebank's syntactic assignments [15], principle (i) corresponds to syntactic chains (i-a) and (i-b) as shown below:

(i-a) (NP|NP-SBJ <text1> (SBAR|RRC <text2>))

(i-b) (NP|NP-SBJ <text1> (PRN <text2> (S <text3>)))

SBJ, SBAR, RRC, PRN, and S stand for subject (SBJ), subordinate clause and relative clause (SBAR), reduce relative clause (RRC), parenthetical (PRN), and sentence (S) respectively. Syntactic chain (i-a) means a subordinate clause or a reduced relative clause is inside a noun phrase. <text1>, <text2>, and <text3> are the context of a noun phrase. For example, consider the sentence “*The land he owns is very valuable.*” The syntactic chain which represents the noun phrase “*The land he owns*” in the above sentence can be written as (NP The land (SBAR he owns)).

If a clause, which is attached to a noun phrase, is headed by a preposition, then the syntactic chain of the noun phrase that corresponds to principle (i) is:

(i-c) (NP|NP-SBJ <text1> (PP <text2> (S|VP <text3>)))

In chain (i-c), PP stands for prepositional phrase. According to principle (i), <text2> in syntactic chain (i-a), and <text2> combining with <text3> in syntactic chains (i-b) and (i-c) are recognised as embedded units. To simplify syntactic chains (i-b) and (i-c), the system creates two labels named PRS (parenthetical-sentence) and PS (prepositional-sentence). These two labels are described respectively in (i-d) and (i-e) below:

(i-d) (PRN <text2> (S <text3>)) → (PRS <text2-3>)

(i-e) (PP <text2> (S|VP <text3>)) → (PS <text2-3>)

“→” can be interpreted as “convert to”. <text2-3> is the concatenated string of <text2> and <text3>. By using syntactic chains (i-d) and (i-e), syntactic chains (i-a) to (i-c) can be grouped into one syntactic chain as follow:

(i-a') (NP|NP-SBJ <text1> (SBAR|RRC|PS|PRS <text2'>))

It should be noted that <text2'> in (i-a') is <text2-3> in (i-d) and (i-e). Due to space constraint, we only represent syntactic chains of the segmentation principles (ii), (iii), and (iv). In the syntactic chains corresponding to principles (ii), (iii), and (iv) as shown below, Sx stands for basic clause types such as subordinate clause and relative clause (SBAR), participial clause (S-ADV),... ”And|but|or...” stands for a conjunction such as “and”, or “but”, or “or”.

The syntactic chain of principle (ii) is:

(ii-a) (Sx <text1> (Sx <text2>) and|but|or... (Sx <text3>))

The syntactic chain of principle (iii) is:

(iii-a) (VP (VP <text1>) and|but|or... (VP|Sx|RRC|PPS <text2>))

The syntactic chains of principle (iv) is:

(iv-a) (S (NP-SBJ <text1>) (VP <text2> (SBAR <text3>)))

(iv-b) (S (NP-SBJ <text1>) (VP <text2> (SBAR <text3>) and|but|or... (SBAR <text4>)))

<text1> in (iv-a) and (iv-b) are not the pronoun “it”.

(iv-c) (Sx (Sx <text1>) , (NP-SBJ <text2>) (VP <text3>))

³ The sentence's pauses can be recognised by a syntactic parser. In this experiment, information about sentence's pauses is in parsed documents of the Penn Treebank.

(iv-d) (Sx (Sx <text1>) , (VP <text2>) (NP-SBJ <text3>))
(iv-e) (Sx (NP-SBJ <text1>) , (Sx <text2>) , (VP <text3>))
(iv-f) (Sx (VP <text1>) (NP-SBJ <text2>) , (Sx <text3>))
<text3> in (iv-c), <text2> in (iv-d), <text3> in (iv-e), and <text1> in (iv-f) are reported verbs or cognitive verbs.

2.2 Segmentation Algorithm

The input to this algorithm is the syntactic string of a sentence, in which <text> is replaced by a token #x,y# (where x,y is the begin and end position of <text> in the sentence being analysed). Each token of the syntactic string of the sentence is separated by a space. For example, the syntactic string of the sentence

(5) “The book I read yesterday is interesting.”
is:
(5a) ((S (NP-SBJ (NP The book) (SBAR I read yesterday)) (VP is (ADJP interesting))))

The input to the segmentation algorithm in this case is:
(5b) ((S (NP-SBJ (NP #0,7#) (SBAR #9,24#))) (VP #26,27# (ADJP #29,39#))) .)

The segmentation algorithm uses a stack to store tokens of the syntactic string during the reading process. It pushes and pops tokens onto and off the stack in order to analyse them. The algorithm ends when the syntactic string is reduced to the string “((S #x,y#) .)”. The steps of the algorithm are described below:

1. Read characters in the input string from left to right and put them onto a stack, until a space is found.

2. Repeat Step 1 until two consecutive close brackets are found on the top of the stack.
3. Pop off strings from the top of the stack into a separate string called “*compared string*” until the number of open brackets and the number of close brackets in the *compared string* are equal.
4. Compare the *compared string* with the sample syntactic strings (e.g., the syntactic string (a’)) to check whether they match or not.
 - 4a. If they match, split the text corresponding to the *compared string* based on the segmentation principles. Store the information about the split text in the system. Go to Step 5.
 - 4b. If they do not match, go to Step 5.
5. Encode the *compared string* as a position tag #x,y# and push it back onto the stack with its syntactic information.
6. Repeat Step 1 to Step 5 until the input string is empty and the stack contains the following tokens, considering from the bottom of the stack: “(“, “(“, “S”, “#x,y#”, “)”, “.”, “)”. ”.

Table 1 represents the segmentation progress of sentence (5). Due to space constraints, some steps of the segmentation process are skipped.

The output of the segmentation algorithm for sentence (5) is two segments, “*The book*” and “*I read yesterday*”, which contribute to one relation. The text “*is interesting*” is not in any text spans of the output. Another procedure, which is called the post process, will be called after the segmentation algorithm in order to deal with this problem. This procedure is described in Section 2.3.

Stack (Top of stack) \rightleftarrows	Input string	Compared string	Operations
	((S (NP-SBJ (NP #0,7#) (SBAR #9,24#))) (VP #26,27# (ADJP #29,39#))) .)		Pushing “(” onto the stack
((S (NP-SBJ (NP #0,7#) (SBAR #9,24#))) (VP #26,27# (ADJP #29,39#)) .)		Pushing “(” onto the stack
((S (NP-SBJ (NP #0,7#) (SBAR #9,24#)) (VP #26,27# (ADJP #29,39#)) .)		Pushing “S” onto the stack
. . . .			
((S (NP-SBJ (NP #0,7#) (SBAR #9,24#))) (VP #26,27# (ADJP #29,39#)) .)			Popping off the strings on top of the stack, generating a compared string
((S (NP-SBJ (NP #0,7#) (SBAR #9,24#))) (VP #26,27# (ADJP #29,39#)) .)		(NP-SBJ (NP #0,7#) (SBAR #9,24#))	Mapping principle 1, splitting text (creating discourse segments), encoding the compared string, pushing it back onto the stack
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#)) .)			Pushing “(” onto the stack
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#)) .)			Pushing “VP” onto the stack
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#)) .)			Pushing “#26,27#” onto the stack
. . . .			
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#)) .)			Popping off the strings on top of the stack, generating a compared string
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#)) .)		(VP #26,27# (ADJP #29,39#))	No principle satisfies, encoding the compared string, pushing it back onto the stack
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#)) .)			Pushing “)” onto the stack
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#)) .)			Popping off the strings on top of the stack, generating a compared string
((S (NP-SBJ #0,24#) (VP #26,27# (ADJP #29,39#))	No principle satisfies, encoding the compared string, pushing it back onto the stack
((S #0,39#) .)			Pushing “.” onto the stack
((S #0,39#) .)			Pushing “)” onto the stack
((S #0,39#) .)			STOP

Table 1. Progress of Segmenting Sentence (5) Using Syntactic Information

a new EDU is created from the beginning position of the cue phrase to the end boundary of the noun phrase. However, this action may create incorrect results:

- (10) [In 1988, Kidder eked out a \$46 million profit, *mainly*][because of severe cost cutting.]

The correct segmentation for the sentence given in example (10) is generated by Step 2, and is given in example (11) below:

- (11) [In 1988, Kidder eked out a \$46 million Profit,][*mainly* because of severe cost cutting.]

Such a situation happens when an adverb stands before the cue phrases. Step 2 deals with such cases, by first detecting the noun phrase, which will be an EDU, and then checking for the appearance of adverbs before a strong cue phrase. If an adverb is found, the new EDU is recognized from the beginning position of the adverb to the end boundary of the noun phrase. Otherwise, the new EDU is split from the beginning position of the cue phrase to the end boundary of the noun phrase, for example:

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

4 Evaluation

Eight documents of the RST Discourse Treebank [16] are used in the experiment. These documents are Wall Street Journal articles from the LDC Treebank [13], which have been annotated with discourse structure by human. The system's input is the corresponding syntactically parsed documents taken from the Penn Treebank. The documents used in this experiment consists of 166 sentences with 3810 words. Most of the sentences are long and complex.

The evaluation is done by comparing the EDUs assigned by the system with the EDUs from the eight RST documents mentioned above. Two EDUs are considered as similar if they have the same boundaries. There are 474 EDUs assigned by the system and 487 EDUs created by human, in which 386 EDUs of these two EDU sets are similar. Thus, there are 88 EDUs created by the system, which are not assigned by human. There are 101 EDUs created by human, which are not assigned by the system.

The standard information retrieval measurements (precision and recall) are used for evaluation. The precision is the proportion of assignments made that were correct. The recall is the proportion of possible assignments that were actually assigned. The precision and the recall of our experiment are:

$$Precision = \frac{386}{386 + 88} = 81.4\% \quad Recall = \frac{386}{386 + 101} = 79.3\%$$

These measurements depend on several factors. The primary factor is the accuracy of syntactic information. The incorrectness syntactic information will decrease the accuracy of the segmentation's result. The syntactic documents from the Penn Treebank, which are used as the input of

our system, also contain analytical errors. Since these errors in the Penn Treebank are rare, this factor does not have a great effect on our system's performance.

The second factor is the difference in human judgements. One person does not always agree with on segmentation [17]. The text in the RST corpus is analysed into very small text spans, which is not how our system segments. For example, consider the segmentation of the following sentence in the RST corpus:

- (13) [Every order shall be presented to the President of the United States;]₇ [and]₈ [before the same shall take effect,]₉ [shall be approved by him,]₁₀ [or]₁₁ [being disapproved by him,]₁₂ [shall be repassed by two-thirds of the Senate and House of Representatives.]₁₃

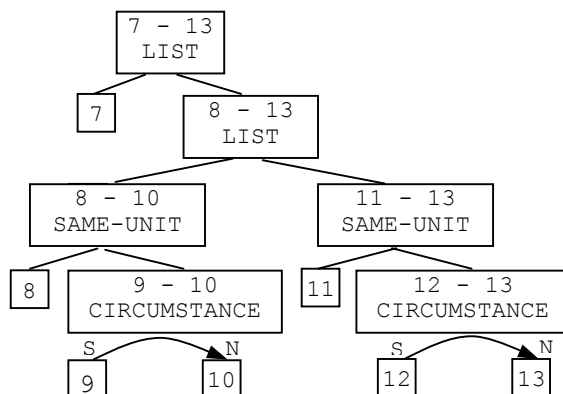


Fig. 3. Discourse Structure of Example (13), Getting from The RST Discourse Corpus⁶

The sentence in example (13) is treated differently by our system, which is shown in example (14):

- (14) [Every order shall be presented to the President of the United States;]₁₄ [and before the same shall take effect,]₁₅ [shall be approved by him,]₁₆ [or being disapproved by him,]₁₇ [shall be repassed by two-thirds of the Senate and House of Representatives.]₁₈

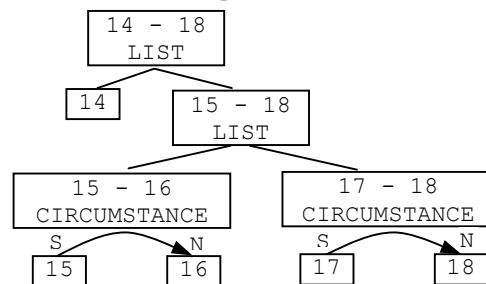


Fig. 4. Discourse Structure of Example (14), Generating by Our System

Over-segmentation is prevented as much as possible in our system because it makes discourse analysis more complicated. The appearance of new discourse units not only affects the EDUs next to them, but also the EDUs in other parts of the text. Since the merging of discourse conjunctions with their clauses does not change the general

⁶ All relation names mentioned in this paper are aiming at making discourse structures clearer. Recognising discourse relations is not in this paper's scope.

meaning of this discourse structure, we analyse the sentence in a different way than that in the RST corpus. This treatment causes some difference between the output of our system with the data from the RST corpus.

As discussed above, incorrect syntactic information and the disagreement in human judgements reduce the system's performance. We accept this reduction because not all discourse structures in the RST corpus are absolutely correct. Several discourse segments in the RST corpus are not accepted by other researchers.

Since researchers are still not certain about the criteria to indicate the exact discourse segment boundaries, and there is no standard benchmark, it is difficult to compare one researcher's result with others. Nonetheless, Okumura and Honda [6] carried out experiments on three texts, which were from exam questions in Japanese. The average precision and recall rates of that experiment were 25% and 52% respectively. The best precision and recall in the series of Passonneau and Litman's experiment [7], which used machine learning approach, were 95% and 53% respectively. Marcu [18] carried out experiments on a corpus of 90 discourse trees, which were built manually from the text in the Message Understanding Conference (MUC) coreference corpus, the Wall Street Journal (WSJ) corpus, and Brown corpus. If the system was trained in all corpora, the precision and recall for testing on WSJ corpus were 79.6% and 25.1%. These values are lower than our system. The precision and recall for MUC corpus were 96.9% and 75.4%; those of Brown corpus were 80.3% and 44.2% respectively. Although several results reported in [7] and [18] are higher than our result, the efficiency of these systems should not be judged purely on these numbers since they depend on other factors such as the size of training corpora, the corpora's domains, and the accuracy of human annotation. Meanwhile, the performance of our system is acceptable because our system does not need any training.

Our system's performance is promising when compared with the systems mentioned above and with other discourse segmentation systems known to us. However, more experimenting using a larger corpus is needed in order to get a more reliable evaluation.

5 Conclusion and Future Work

In this paper, we have presented a discourse segmentation method based on syntax and cue phrases. The discourse segmenter consists of two modules. Firstly, text is split based on syntactic information, aiming at receiving discourse units with independent functional integrity. Secondly, noun phrases that have the role of EDUs are recognised by detecting strong cue phrases from text.

Our preliminary experiment shows that this method attains promising results without any training. The experimental result is encouraging in comparison with existing segmentation methods. However, the system's performance

can still be improved by the following ways: investigating a method to reduce the effect of syntactic information; and refining the rules for segmentation by syntax and for post processing. We leave these tasks for future work. Future work also includes integrating a syntactic parser with the discourse segmenter. Since there are many advanced syntactic parsers currently available, this problem can be easily solved.

A discourse parser cannot provide good results without accurate discourse segmentation. Therefore, this research is important in building discourse analysing systems, which have a wide range of applications including text summarisation.

References

- [1] Mann, W. C. and Thompson, S. A., Rhetorical Structure Theory: Toward a Functional Theory of Text Organization, *Text*, 8, 1988, 243-281.
- [2] Grosz, B.J. and Sidner C.L., Attention, intentions and the structure of discourse, *Computational Linguistics*, 12, 1986, 175-204.
- [3] Marcu, D., *The Rhetorical Parsing, Summarisation, and Generation of Natural Language Texts* (Ph.D. Thesis: Department of Computer Science, University of Toronto, 1997).
- [4] Morris, J., & Hirst, G., Lexical Cohesion Computed by Thesaural Relations as an Indicator of the Structure of the Text, *Computational Linguistics*, 17, 1991, 21-28.
- [5] Kozima, H., *Computing lexical cohesion as a tool for text analysis* (Ph.D. Thesis: Graduate School of Electro-Communications, University of Electro-Communications, 1994).
- [6] Okumura, M. and Honda, T., Word Sense Disambiguation and Text Segmentation Based on Lexical Cohesion, *Proc. of the 15th Conf. on Computational Linguistics (COLING-94)*, 2, 1994, 755-761.
- [7] Passonneau, R. J. and Litman, D. J., Discourse Segmentation by Human and Automated Means, *Computational Linguistics*, 23(1), 1997, 103-139.
- [8] Forbes, K. and Miltsakaki, E., Empirical Studies of Centering Shifts and Cue Phrases as Embedded Segment Boundary Markers, *Penn Working Papers in Linguistics*, 7(2), 2002, 39-57.
- [9] Batliner, A., Kompe, R., Kießling, A., Niemann, H., and Nöth, E., Syntactic-Prosodic Labeling Of Large Spontaneous Speech Databases, *Proc. of ICSLP, USA*, 1996.
- [10] Corston-Oliver, S., *Computing Representations of the Structure of Written Discourse* (Ph.D. Thesis: University of California, Santa Barbara, CA, U.S.A., 1998).
- [11] Webber, B. L., Structure and ostension in the interpretation of discourse deixis, *Language and Cognitive Processes*, 6(2), 1991, 107-135.
- [12] Carlson, L., Marcu, D., and Okurowski, M. E., RST Discourse Treebank, *LDC*, 2002.
- [13] Marcus, M. P., Santorini, B. and Marcinkiewicz, M.A., Penn Treebank II, *LDC*, 1995.
- [14] Carlson, L. and Marcu, D., Discourse Tagging Manual, *ISI Tech Report*, ISI-TR-545, 2001.
- [15] Bies, A. et al., Bracketing Guidelines for Treebank II Style, *Penn Treebank Project*, 1995.
- [16] Carlson, L., Marcu, D., and Okurowski, M. E., *Eight documents of the RST Discourse Treebank* (from <http://www.isi.edu/%7Emarcu/>, 2002)
- [17] Litman, D. J. and Passonneau, R. J., Intention-based segmentation: Human reliability and correlation with linguistic cues, *Proc. of the 31st Annual Meeting of the Association for Computational Linguistics*, 1993, 148-155.
- [18] Marcu, D., A Decision-Based Approach to Rhetorical Parsing, *The 37th Annual Meeting of the Association for Computational Linguistics (ACL)*, Maryland, 1999, 365-372.