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 English written text into Elementary Discourse Units (EDUs) 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 can give promising result in comparison with existing research in discourse segmentation.

Keywords: 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 (Mann and Thompson, 1988; Gross and Sidner, 1986). According to Mann and Thompson (1988), all discourse units should have independent functional integrity, such as independent clauses. These units are called elementary discourse units (Marcu, 1997).

It has been claimed that discourse has been segmented using disparate phenomena: lexical cohesion (Morris and Hirst, 1991; Kozima, 1994; Okumura and Honda, 1994), discourse cues (Grosz and Sidner, 1986; Passonneau and Litman, 1997; Marcu, 1997; Forbes and Milsakaki, 2002), and syntactic information (Batliner et al., 1996; Corston-Oliver, 1998). However, the criteria to indicate the exact discourse segment boundaries are still not certain. The three approaches mentioned above are further discussed below.

The weakness of the lexical cohesion approach is that it can not guarantee independent discourse units, which is the essential condition for discourse segmentation. Discourse cues, such as cue phrases, pauses, and referential identities (Webber, 1991; Marcu, 1997) can be a solution for this problem. Marcu (1997)’s shallow analyser splits text into EDUs by mapping cue phrases and punctuation mark. However, this approach cannot correctly identify boundaries in complex sentences, which do not have any lexical discourse cues.

Passonneau and Litman (1997) 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 use 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. Corston-Oliver (1998) used a syntactic approach. He defined a rule set for discourse segmentation based 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.

To solve 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 (Mann and Thompson, 1988), the text is first split 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, discourse segmentation by syntax, is described in Section 2. Section 3 analyses the post processing of the output of Step 1. Discourse segmentation by cue phrases is represented in Section 4. In Section 5, the system is evaluated. Finally, we present our conclusions and a discussion of possible future work in Section 6.

2 Step 1: Discourse Segmentation by Syntax

The module takes parsed documents from Penn Treebank as its input. One sentence is analyzed at each round of the segmentation process. This module not only splits sentences into clauses, but also provides some initial information about discourse relations between two EDUs, such as which two EDUs in a sentence may have a discourse relation, and which EDU plays the nucleus/satellite role.

\[\text{1 For further information on “EDUs”, see (Marcu, 1997).}\]

\[\text{2 The biggest discourse corpus nowadays is the RST Discourse Treebank from LDC (Linguistic Data Consortium), with 385 Wall Street Journal articles.}\]

\[\text{3 The sentence’s pauses can be recognised by a syntactic parser. In this experiment, information about sentence’s pauses is in parsed documents of the Penn Treebank.}\]
2.1 Segmentation Principles

The principles for segmenting text into discourse units in this step are based on the syntactic relations between text. The main principles 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.)]

Principle (i) corresponds to syntactic chains (a) and (b) as shown below:

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

(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 (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, the syntactic chain which represents the noun phrase “The land he owns” in the sentence “The land he owns is very valuable.” can be written as (NP The land [SBAR he owns]).

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

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

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

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

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

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

(a’) (NP[NP-SBJ <text1> (SBAR[RRC|PS|PRS <text2’>)]})

It should be noted that <text2’> in (a’) is <text2-3> in (d) and (e). Due to space constraint, the syntactic chains that represent segmentation principles (ii), (iii), and (iv) are not described in this paper.

The segmentation algorithm will be described in the rest of this Section. The input of this algorithm is the syntactic string of a sentence, in which <text> is replaced by a token #x,y# (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 of the segmentation algorithm in this case will be:

(5b) (( S ( NP-SBJ ( NP #0,7# ) ( SBAR #9,24# ) ) (VP #26,27# (ADJP #29,39# ) ) ) . )

2.2 Segmentation Algorithm

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

1. Reading characters in the input string from left to right until a space is found. Those characters are put into the stack.

2. Continuing Step 1 until two intermediate close brackets are found on the top of the stack.

3. Popping out strings on the top of the stack into a separate string until the number of open brackets and the number of close brackets in this new string are equal. This string is called a “compared string”.

4. The “compared string” is compared with the sample syntactic strings (e.g., the syntactic string (a’)) to check whether they match or not.

4a. If they match, the text corresponding to the “compared string” is split based on the segmentation principles. The information about the split text is stored in the system. Go to Step 5.

4b. If they do not match, go to Step 5.

5. The text which is pop out from the stack in Step 3 is encoded as a position tag #x,y# and pushed back into the stack with its syntactic information.

6. Repeat Step 1 to Step 6 until the input string is empty and the stack contains the following tokens, counting from the bottom of the stack: “(”, “(“,” “S”, “#x,y#”, “)”, “)”, “)”).
Table 1 represents the segmentation progress of sentence (5). Due to space constrain, some steps of the segmentation process are replaced by "...".

The output of Step 1 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 process will be called after this segmentation process in order to deal with this problem. This process will be described in Section 3.

3 Step 2: Post Processing the Output of Step 1

The purpose of this step is to refine the output of Step 1. There are two situations needing this process. The first situation is that the segmentation of embedded units makes the text fragmented. For example, sentence (5) after being processed by Step 1 will have the structure as shown below:

\[
(1-2) \quad \text{SAME-UNIT} \quad N \quad S
\]

In fact, the text "is interesting" cannot be a single EDU because it does not have independent functional integrity.

Meanwhile, the embedded clause "I read yesterday" provides additional information for the noun phrase "the book". "The book" is the nucleus (N) (the most important part in the relation); "I read yesterday" is the satellite (S) in the relation. In this case, a relation called SAME-UNIT\(^5\) is created between "The book I read yesterday" and "is interesting". Both text spans "The book I read yesterday" and "is interesting" have an equal important role in contributing to the sentence "The book I read yesterday is interesting". Therefore, both of them are nucleus in SAME-UNIT relation.

\[
\text{Table 1. Progress of segmenting sentence (5) using syntactic information}
\]

<table>
<thead>
<tr>
<th>Stack (Top of stack)</th>
<th>Input string</th>
<th>Compared string</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>(( ( S ( NP-SBJ ( NP #0,7# ) ( SBAR #9,24# ) ) ( VP #26,27# ( ADJP #29,39# ) ) ) . ) )</td>
<td>Pushing &quot;(&quot; to the stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( ( S ( NP-SBJ ( NP #0,7# ) ( SBAR #9,24# ) ) ( VP #26,27# ( ADJP #29,39# ) ) ) . )</td>
<td>Pushing &quot;)&quot; to the stack</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( ( S ( NP-SBJ #0,39# ) ) )</td>
<td>Pushing &quot;,&quot; to the stack</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{Fig. 1. Discourse structure of example (6)}
\]

\[^4\text{UNKNOWN text span specifies the text fragment after segmenting the sentence. It is not a discourse relation.}\]

\[^5\text{SAME-UNIT means that two text spans are on the same discourse unit (Marcu, 1997).}\]
The post process’s operation depends on the position of the embedded unit. When the satellite of a relation is near an UNKNOWN text span, a SAME-UNIT relation is assigned between the UNKNOWN text span and the text span that contains the nucleus and satellite. Otherwise, when the nucleus of a relation is adjacent to an UNKNOWN text span, the UNKNOWN text span is merged with the nucleus, as in example (7) below.

(7) Mr. Silas Cathcart built a shopping mall on some land he owns.

The segmentation by syntax algorithm finds two segments in the sentence (7), “some land” and “he owns”, but the actual segments should be “Mr. Silas Cathcart built a shopping mall on some land” and “he owns”.

Fig. 2 presents the discourse structure of the sentence in example (7). “Mr. Silas Cathcart built a shopping mall on some land” is the nucleus (N); “he owns” is the satellite (S) in the relation. The dotted line shows the syntactic relation between “some land” and “he owns”. The solid line shows a discourse relation between the two actual discourse units, after the sentence has been processed by Step 2.

![Fig. 2. Discourse structure of example (7)](image)

The second situation needed post processing involves the placement of adverbs in EDUs. Some adverbs, which should stand at the beginning of the right clause, are put at the end of the left clause by the process in Step 1. This situation is detected and corrected by the procedure in Step 2. Examples (8) and (9) show such a situation. The clause “they did not have enough people” is split from the sentence “They had to give up that campaign, mainly because they did not have enough people” by syntactic information in Step 1. However, the accurate segmentation in this case should be “mainly because they did not have enough people”, not “they did not have enough people”. After undergoing the process in Step 2, the boundary created by Step 1 is moved backward to the position between the comma and the two adverbs “mainly because”, as shown in example (9).

(8) [They had to give up that campaign, mainly because][they did not have enough people.]

(9) [They had to give up that campaign, mainly because they did not have enough people.]

The input to the post processing procedure is the output of Step 1. The output of Step 2 is the discourse segments after refining boundaries.

4 Step 3: Discourse Segmentation by Cue Phrases

Several noun phrases are considered as EDUs when they are accompanied by strong cue phrases. These cases cannot be recognized by syntactic information. Therefore, another segmentation process is integrated into the system to deal with such cases. This process finds strong cue phrases from the output of Step 2. When a strong cue phrase is found, the algorithm seeks the end boundary of the noun phrase. These end boundaries can be terminal characters such as a comma, semicolon, or full stop. Normally, a new EDU is created from the beginning position of the cue phrase to the end boundary of the phrase. However, this action may create incorrect results, as shown in example (10) below:

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

The correct segmentation for example (11), which is generated by the algorithm in Step 3, is:

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

Such a situation happens when the adverb stands before the cue phrases. The algorithm in Step 3 deals with such cases, by first detecting the noun phrase which will be an EDU, and then checking for the appearance of adverbs before the 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, such as example (12) below:

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

5 Evaluation

Eight documents of RST Discourse Treebank (2002) which are provided on Marcu’s web site (http://www.isi.edu/%7Emarcu/) are used in the experiment. These documents are Wall Street Journal articles from the LDC Treebank, which have been annotated with discourse structure by human judges. The system’s input is the corresponding syntactically parsed documents taken from the Penn Treebank. The documents used in this experiment consist of 166 sentences with 3810 words. Most of the sentences are long and complex.

The experimental result is shown in Table 2. The evaluation is done by comparing the EDUs generated (algorithm’s assignments) by the system with the leaves of the
discourse tree from the RST documents mentioned above (human judges). The total EDUs created by the algorithm, by human, and the EDUs shared by them are counted. Two EDUs are equal if they have the same boundaries.

<table>
<thead>
<tr>
<th></th>
<th>Human judges</th>
<th>Algorithm’s assignments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDU</td>
<td>386</td>
<td>88</td>
</tr>
<tr>
<td>¬EDU</td>
<td>101</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Performance of the discourse segmenter

The standard information retrieval measurements (precision and recall) are used for evaluating. 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:

\[
\text{Precision} = \frac{386}{386 + 88} = 81.43\% \\
\text{Recall} = \frac{386}{386 + 101} = 79.26\%
\]

These measurements depend on several factors. The primary factor is the accuracy of syntactic information. Incorrect of syntactic information will decrease the accuracy of the segmentation’s result. The syntactic documents from Penn Treebank, which are used as the input of our system, also contain analytical errors. Since these errors in 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 other people on segmentation (Passonneau and Litman, 1993). 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

The sentence in example (13) is treated differently 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 meaning of this discourse structure, we analyse the sentence in a different way than that in the RST corpus.

(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. 3. Discourse structure of example (13), getting from RST discourse corpus

Fig. 4. Discourse structure of example (14), generating by our system

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 people 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 (1994) carried out experiments on three texts, which are 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 (1997)’s experiment were 95% and 53% respectively. Marcu (1999) 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 making discourse structures clearer. Recognising discourse relations is not in this paper’s scope.

---

6 All relation names mentioned in this paper are aiming at
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 (Passonneau and Litman, 1997) and (Marcu, 1999) are higher than our result, they depend on training corpora. Meanwhile, the performance of our system does not depend on discourse corpora since 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 more reliable evaluation.

6 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 three processes. Firstly, text is split based on syntactic information, aiming at receiving discourse units with independent functional integrity. Secondly, the output of the previous process is modified to get the exact discourse segment boundaries. Finally, noun phrases that have the role of EDUs are recognised by detecting strong cue phrases from text.

Our preliminary experiment shows that this method can attain 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; 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