A surface-based approach to identifying discourse markers and elementary textual units in unrestricted texts

Daniel Marcu
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292–6695
marcu@isi.edu

Abstract

I present a surface-based algorithm that employs knowledge of cue phrase usages in order to determine automatically clause boundaries and discourse markers in unrestricted natural language texts. The knowledge was derived from a comprehensive corpus analysis.

1 Motivation

The automatic identification of discourse segments and discourse markers in unrestricted texts is crucial for solving many outstanding problems in natural language processing, which range from syntactic and semantic analysis, to anaphora resolution and text summarization. Most of the algorithmic research in discourse segmentation focused on segments of coarse granularity (Grosz and Hirschberg, 1992; Hirschberg and Litman, 1993; Passonneau and Litman, 1997; Hearst, 1997; Yaari, 1997). These segments were defined intentionally in terms of Grosz and Sidner's theory (1986) or in terms of an intuitive notion of "topic".

However, in case of applications such as anaphora resolution, discourse parsing, and text summarization, even sentences might prove to be too large discourse segments. For example, if we are to derive the discourse structure of texts using an RST-like representation (Mann and Thompson, 1988), we will need to determine the elementary textual units that contribute rhetorically to the understanding of those texts; usually, these units are clause-like units. Also, if we want to select the most important parts of a text, sentences might prove again to be too large segments (Marcu, 1997a; Teufel and Moens, 1998): in some cases, only one of the clauses that make up a sentence should be selected for summarization.

In this paper, I present a surface-based algorithm that uses cue phrases (connectives) in order to determine not only the elementary textual units of text but also the phrases that have a discourse function. The algorithm is empirically grounded in an extensive corpus analysis of cue phrases and is consistent with the psycholinguistic position advocated by Caron (1997, p. 70). Caron argues that "rather than conveying information about states of things, connectives can be conceived as procedural instructions for constructing a semantic representation". Among the three procedural functions of segmentation, integration, and inference that are used by Noordman and Vonk (1997) in order to study the role of connectives, I will concentrate here primarily on the first.[1]

2 A corpus analysis of cue phrases

I used previous work on coherence and cohesion to create an initial set of more than 450 potential discourse markers (cue phrases). For each cue phrase, I then used an automatic procedure that extracted from the Brown corpus a random set of text fragments that each contained that cue. On average, I selected approximately 17 text fragments per cue phrase, having few texts for the cue phrases that do not occur very often in the corpus and up to 60 for cue phrases, such as *and*, that I considered to be highly ambiguous. Overall, I randomly selected more than 7600 texts. Marcu (1997b) lists all cue phrases that were used to extract text fragments from the Brown corpus, the number of occurrences of each cue phrase in the corpus, and the number of text fragments that were randomly extracted for each cue phrase.

All the text fragments associated with a potential discourse marker were paired with a set of slots in which I described, among other features, the following: 1. The orthographic environment that characterized the usage of the potential discourse marker. This included occurrences of periods, commas, colons, semicolons, etc. 2. The type of usage: *Sentential, Discourse*, or *Pragmatic*. 3. The

[1] Marcu (1997b) studies the other two functions as well.

position of the marker in the textual unit to which it belonged: *Beginning, Medial,* or *End.* 4. The right boundary of the textual unit associated with the marker. 5. A name of an "action" that can be used by a shallow analyzer in order to determine the elementary units of a text. The shallow analyzer assumes that text is processed in a left-to-right fashion and that a set of flags monitors the segmentation process. Whenever a cue phrase is detected, the shallow analyzer executes an action from a predetermined set, whose effect is one of the following: create an elementary textual unit boundary in the input text stream; or set a flag. Later, if certain conditions are satisfied, the flag setting may lead to the creation of a textual unit boundary. Since a discussion of the actions is meaningless in isolation, I will provide it in conjunction with the clause-like unit boundary and marker-identification algorithm.

The algorithm described in this paper relies on the results derived from the analysis of 2200 of the 7600 text fragments and on the intuitions developed during the analysis.

3 The clause-like unit boundary and marker-identification algorithm

3.1 Determining the potential discourse markers

The corpus analysis discussed above provides information about the orthographic environment of cue phrases and the function that they have in texts. A cue phrase was assigned a *sentential* role, when it had no function in structuring the discourse; a *discourse* role, when it signalled a discourse relation between two textual units; or a *pragmatic* role, when it signalled a relationship between a linguistic or nonlinguistic construct that pertained to the unit in which the cue phrase occurred and the beliefs, plans, intentions, and/or communicative goals of the speaker, hearer, or some character depicted in the text. In this case, the beliefs, plans, etc., did not have to be explicitly stated in discourse; rather, it was the role of the cue phrase to help the reader infer them.[2]

Different orthographic environments often correlate with different discourse functions. For example, if the cue phrase *Besides* occurs at the beginning of a sentence and is not followed by a comma, as in text (1), it usually signals a rhetorical relation that holds between the clause-like unit that contains it and the clause that comes after. However, if the same cue phrase occurs at the beginning of a sentence and is immediately followed by a comma, as in text (2), it usually signals a rhetorical relation that holds between the sentence to which *Besides* belongs and a textual units that precedes it.

(1) [*Besides* the lack of an adequate ethical dimension to the Governor's case,] [one can ask seriously whether our lead over the Russians in quality and quantity of nuclear weapons is so slight as to make the tests absolutely necessary.]

(2) [For pride's sake, I will not say that the coy and leering vade mecum of those verses insinuated itself into my soul.] [*Besides,* that particular message does no more than weakly echo the roar in all fresh blood.]

I have taken each of the cue phrases in the corpus and evaluated its potential contribution in determining the elementary textual units and discourse function for each orthographic environment that characterized its usage.

I used the cue phrases and the orthographic environments that characterized the cue phrases that played a discourse role in most of the text fragments in the corpus in order to manually develop a set of regular expressions that can be used to recognize potential discourse markers in naturally occurring texts. If a cue phrase had different discourse functions in different orthographic environments, as was the case with *Besides*, I created one regular expression for each function. I ignored the cue phrases that played a sentential role in a majority of the text fragments and the cue phrases for which I was not able to infer straightforward rules that would allow a shallow algorithm to discriminate between their discourse and sentential usages. Because orthographic markers, such as commas, periods, dashes, paragraph breaks, etc., play an important role in the surface-based approach to discourse processing that I present here, I included them in the list of potential discourse markers as well.

3.2 From the corpus analysis to the elementary textual units of a text

During the corpus analysis, I generated a set of eleven actions that constitutes the foundation of an algorithm to determine automatically the elementary units of a text. The algorithm processes a text given as input in a left-to-right fashion and "executes" the

[2] This definition of pragmatic connective was first proposed by Fraser (1996). It should not be confused with the definition proposed by van Dijk (1979), who calls a connective "pragmatic" if it relates two speech acts and not two semantic units.

actions that are associated with each potential discourse marker and each punctuation mark that occurs in the text. Because the algorithm does not use any traditional parsing or tagging techniques, I call it a "shallow analyzer".

The names and the intended semantics of the actions used by the shallow analyzer are:

- Action NOTHING instructs the shallow analyzer to treat the cue phrase under consideration as a simple word. That is, no textual unit boundary is normally set when a cue phrase associated with such an action is processed. For example, the action associated with the cue phrase *accordingly* is NOTHING.
- Action NORMAL instructs the analyzer to insert a textual boundary immediately before the occurrence of the marker. Textual boundaries correspond to elementary unit breaks.
- Action COMMA instructs the analyzer to insert a textual boundary immediately after the occurrence of the first comma in the input stream. If the first comma is followed by an *and* or an *or*, the textual boundary is set after the occurrence of the next comma. If no comma is found before the end of the sentence, a textual boundary is created at the end of the sentence.
- Action NORMAL_THEN_COMMA instructs the analyzer to insert a textual boundary immediately before the occurrence of the marker and another textual boundary immediately after the occurrence of the first comma in the input stream. As in the case of the action COMMA, if the first comma is followed by an *and* or an *or*, the textual boundary is set after the occurrence of the next comma. If no comma is found before the end of the sentence, a textual boundary is created at the end of the sentence.
- Action END instructs the analyzer to insert a textual boundary immediately after the cue phrase.
- Action MATCH_PAREN instructs the analyzer to insert textual boundaries both before the occurrence of the open parenthesis that is normally characterized by such an action, and after the closed parenthesis that follows it.
- Action COMMA_PAREN instructs the analyzer to insert textual boundaries both before the cue phrase and after the occurrence of the next comma in the input stream.
- Action MATCH_DASH instructs the analyzer to insert a textual boundary before the occurrence of the cue phrase. The cue phrase is usually a dash. The action also instructs the analyzer to insert a textual boundary after the next dash in the text. If such a dash does not exist, the textual boundary is inserted at the end of the sentence.

The preceding three actions, MATCH_PAREN, COMMA_PAREN, and MATCH_DASH, are usually used for determining the boundaries of parenthetical units. These units, such as those shown in italics in (3) and (4) below, are related only to the larger units that they belong to or to the units that immediately precede them.

(3) [With its distant orbit {— *50 percent farther from the sun than the Earth* —} and slim atmospheric blanket,] [Mars experiences frigid weather conditions.]

(4) [Yet, even on the summer pole, {*where the sun remains in the sky all day long,*} temperatures never warm enough to melt frozen water.]

Because the deletion of parenthetical units does not affect the readability of a text, in the algorithm that I present here I do not assign them an elementary unit status. Instead, I will only determine the boundaries of parenthetical units and record, for each elementary unit, the set of parenthetical units that belong to it.

- Actions SET_AND (SET_OR) instructs the analyzer to store the information that the input stream contains the lexeme *and* (*or*).
- Action DUAL instructs the analyzer to insert a textual boundary immediately before the cue phrase under consideration if there is no other cue phrase that immediately precedes it. If there exists such a cue phrase, the analyzer will behave as in the case of the action COMMA. The action DUAL is usually associated with cue phrases that can introduce some expectations about the discourse (Cristea and Webber, 1997). For example, the cue phrase *although* in text (5) signals a rhetorical relation of CONCESSION between the clause to which it belongs and the previous clause. However, in text (6), where *although* is preceded by an *and*, it signals a rhetorical relation of CONCESSION between the clause to which it belongs and the next clause in the text.

  (5) [I went to the theater] [*although* I had a terrible headache.]

(6) [The trip was fun,] [*and although* we were badly bitten by blackflies,] [I do not regret it.]

3.3 The clause-like unit and discourse-marker identification algorithm

On the basis of the information derived from the corpus, I have designed an algorithm that identifies elementary textual unit boundaries in sentences and cue phrases that have a discourse function. Figure 1 shows only its skeleton and focuses on the variables and steps that are used in order to determine the elementary units. Due to space constraints, the steps that assert the discourse function of a marker are not shown; however, these steps are mentioned in the discussion of the algorithm that is given below. Marcu (1997b) provides a full description of the algorithm.

The algorithm takes as input a sentence S and the array markers[n] of cue phrases (potential discourse markers) that occur in that sentence; the array is produced by a trivial algorithm that recognizes regular expressions (see section 3.1). Each element in markers[n] is characterized by a feature structure with the following entries:

- the action associated with the cue phrase;
- the position in the elementary unit of the cue phrase;
- a flag *has_discourse_function* that is initially set to "no".

The clause-like unit and discourse-marker identification algorithm traverses the array of cue phrases left-to-right (see the loop between lines 2 and 20) and identifies the elementary textual units in the sentence on the basis of the types of the markers that it processes. Crucial to the algorithm is the variable "status", which records the set of markers that have been processed earlier and that may still influence the identification of clause and parenthetical unit boundaries.

The clause-like unit identification algorithm has two main parts: lines 10–20 concern actions that are executed when the "status" variable is NIL. These actions can insert textual unit boundaries or modify the value of the variable "status", thus influencing the processing of further markers. Lines 3–9 concern actions that are executed when the "status" variable is not NIL. We discuss now in turn each of these actions.

Lines 3–4 of the algorithm treat parenthetical information. Once an open parenthesis, a dash, or a discourse marker whose associated action is COMMA_PAREN has been identified, the algorithm ignores all other potential discourse markers until the element that closes the parenthetical unit is processed. Hence, the algorithm searches for the first closed parenthesis, dash, or comma, ignoring all other markers on the way. Obviously, this implementation does not assign a discourse usage to discourse markers that are used *within* a span that is parenthetic. However, this choice is consistent with the decision discussed in section 3.2, to assign parenthetical information no elementary textual unit status. Because of this, the text shown in italics in text (7), for example, is treated as a single parenthetical unit, which is subordinated to "Yet, even on the summer pole, temperatures never warm enough to melt frozen water". In dealing with parenthetical units, the algorithm avoids setting boundaries in cases in which the first comma that comes after a COMMA_PAREN marker is immediately followed by an *or* or *and*. As example (7) shows, taking the first comma as boundary of the parenthetical unit would be inappropriate.

(7) [Yet, even on the summer pole, {*where the sun remains in the sky all day long, and where winds are not as strong as at the Equator,*} temperatures never warm enough to melt frozen water.]

Obviously, one can easily find counterexamples to this rule (and to other rules that are employed by the algorithm). For example, the clause-like unit and discourse-marker identification algorithm will produce erroneous results when it processes the sentence shown in (8) below.

(8) [I gave John a boat,] [which he liked, and a duck,] [which he didn't.]

Nevertheless, the evaluation results discussed in section 4 show that the algorithm produces correct results in the majority of the cases.

If the "status" variable contains the action COMMA, the occurrence of the first comma that is not adjacent to an *and* or *or* marker determines the identification of a new elementary unit (see lines 5–7 in figure 1).

Usually, the discourse role of the cue phrases *and* and *or* is ignored because the surface-form algorithm that we propose is unable to distinguish accurately enough between their discourse and sentential usages. However, lines 8–9 of the algorithm concern cases in which their discourse function can be unambiguously determined. For example, in our

```
Input:    A sentence S.
          The array of $n$ potential discourse markers markers[$n$] that occur in S.
Output:   The clause-like units, parenthetical units, and discourse markers of S.

1.  status := NIL; . . . ;
2.  for $i$ from 1 to $n$
3.      if MATCH_PAREN $\in$ status $\vee$ MATCH_DASH $\in$ status $\vee$ COMMA_PAREN $\in$ status
4.        ⟨deal with parenthetical information⟩
5.      if COMMA $\in$ status $\wedge$ markerTextEqual($i$,“,”) $\wedge$
6.          NextAdjacentMarkerIsNotAnd() $\wedge$ NextAdjacentMarkerIsNotOr()
7.        ⟨insert textual boundary after comma⟩
8.      if (SET_AND $\in$ status $\vee$ SET_OR $\in$ status) $\wedge$ markerAdjacent($i-1, i$)
9.        ⟨deal with adjacent markers⟩
10.     switch(getActionType($i$)){
11.         case DUAL: ⟨deal with DUAL markers⟩
12.         case NORMAL: ⟨insert textual boundary before marker⟩
13.         case COMMA: status := status $\cup$ {COMMA};
14.         case NORMAL_THEN_COMMA: ⟨insert textual boundary before marker⟩
15                                  status := status $\cup$ {COMMA};
16.         case NOTHING: ⟨assign discourse usage⟩$\star$
17.         case MATCH_PAREN, COMMA_PAREN, MATCH_DASH: status := status $\cup$ {getActionType($i$)};
18.         case SET_AND, SET_OR: status := status $\cup$ {getActionType($i$)};
19.     }
20. end for
21. finishUpParentheticalsAndClauses();
```

Figure 1: The skeleton of the clause-like unit and discourse-marker identification algorithm

corpus, whenever *and* and *or* immediately preceded the occurrence of other discourse markers (function markerAdjacent($i-1, i$) returns true), they had a discourse function. For example, in sentence (9), *and* acts as an indicator of a JOINT relation between the first two clauses of the text.

(9) [Although the weather on Mars is cold] [*and although* it is very unlikely that water exists,] [scientists have not dismissed yet the possibility of life on the Red Planet.]

If a discourse marker is found that immediately follows the occurrence of an *and* (or an *or*) and if the left boundary of the elementary unit under consideration is found to the left of the *and* (or the *or*), a new elementary unit is identified whose right boundary is just before the *and* (or the *or*). In such a case the *and* (or the *or*) is considered to have a discourse function as well, so the flag *has_discourse_function* is set to “yes”.

If any of the complex conditions in lines 3, 5, or 8 in figure 1 is satisfied, the algorithm not only inserts textual boundaries as discussed above, but it also resets the “status” variable to NIL.

Lines 10–19 of the algorithm concern the cases in which the “status” variable is NIL. If the type of the marker is DUAL, the determination of the textual unit boundaries depends on the marker under scrutiny being adjacent to the marker that precedes it. If it is, the “status” variable is set such that the algorithm will act as in the case of a marker of type COMMA. If the marker under scrutiny is not adjacent to the marker that immediately preceded it, a textual unit boundary is identified. This implementation will modify, for example, the variable “status” to COMMA when processing the marker *although* in example (10), but only insert a textual unit boundary when processing the same marker in example (11). The final textual unit boundaries that are assigned by the algorithm are shown using square brackets.

(10) [John is a nice guy,] [*but although* his colleagues do not pick on him,] [they do not invite him to go camping with them.]

(11) [John is a nice guy,] [*although* he made a couple of nasty remarks last night.]

Line 12 of the algorithm concerns the most frequent marker type. The type NORMAL determines the identification of a new clause-like unit boundary just before the marker under scrutiny. Line 13 concerns the case in which the type of the marker is COMMA. If the marker under scrutiny is adjacent to the previous one, the previous marker is considered to have a discourse function as well. Either case, the "status" variable is updated such that a textual unit boundary will be identified at the first occurrence of a comma. When a marker of type NORMAL_THEN_COMMA is processed, the algorithm identifies a new clause-like unit as in the case of a marker of type NORMAL, and then updates the variable "status" such that a textual unit boundary will be identified at the first occurrence of a comma. In the case a marker of type NOTHING is processed, the only action that might be executed is that of assigning that marker a discourse usage.

Lines 7–8 of the algorithm concern the treatment of markers that introduce expectations with respect to the occurrence of parenthetical units: the effect of processing such markers is that of updating the "status" variable according to the type of the action associated with the marker under scrutiny. The same effect is observed in the cases in which the marker under scrutiny is an *and* or an *or*.

After processing all the markers, it is possible that some text will remain unaccounted for: this text usually occurs between the last marker and the end of the sentence. The procedure "finishUpParentheticalsAndClauses()" in line 21 of figure 1 flushes this text into the last clause-like unit that is under consideration.

4 Evaluation

To evaluate a C++ implementation of the clause-like unit and discourse-marker identification algorithm, I randomly selected three texts, each belonging to a different genre: an expository text of 5036 words from *Scientific American*; a magazine article of 1588 words from *Time*; and a narration of 583 words from the Brown Corpus. No fragment of any of the three texts was used during the corpus analysis. Three independent judges, graduate students in computational linguistics, broke the texts into elementary units. The judges were given no instructions about the criteria that they were to apply in order to determine the clause-like unit boundaries; rather, they were supposed to rely on their intuition and preferred definition of clause. The locations in texts that were labelled as clause-like unit boundaries by at least two of the three judges were considered to be "valid elementary unit boundaries". I used the valid elementary unit boundaries assigned by judges as indicators of discourse usages of cue phrases and I determined manually the cue phrases that signalled a discourse relation. For example, if an *and* was used in a sentence and if the judges agreed that a textual unit boundary existed just before the *and*, I assigned that *and* a discourse usage. Otherwise, I assigned it a sentential usage. Hence, although the corpus analysis was carried out by only one person, the validation of the actions and of the algorithm depicted in figure 1 was carried out against unseen texts, which were manually labelled by multiple subjects.

Once the "gold-standard" textual unit boundaries and discourse markers were manually identified, I applied the algorithm in figure 1 on the same texts. The algorithm found 80.8% of the discourse markers with a precision of 89.5% (see Marcu (1997b) for details), a result that outperforms Hirschberg and Litman's (1993) and its subsequent improvements (Litman, 1996; Siegel and McKeown, 1994).

The algorithm correctly identified 81.3% of the clause-like unit boundaries, with a precision of 90.3%. I am not aware of any surface-form algorithms that achieve similar results. Still, the clause-like unit and discourse-marker identification algorithm has its limitations. These are primarily due to the fact that the algorithm relies entirely on cue phrases and orthographic features that can be detected by shallow methods. For example, such methods are unable to classify correctly the sentential usage of *but* in example (12); as a consequence, the algorithm incorrectly inserts a textual unit boundary before it.

(12) [The U.S. has] [*but* a slight chance to win a medal in Atlanta,] [because the championship eastern European weight-lifting programs have endured in the newly independent countries that survived the fracturing of the Soviet bloc.]

5 Conclusion

In this paper, I have shown how by adopting a procedural view of cue phrases, one can determine automatically the elementary units and discourse markers of texts, with recall and precision figures in the range of 80 and 90% respectively, when compared to humans. The main advantage of the proposed algorithm is its speed: it is linear in the size of the input. It is the purpose of future research to improve the algorithm described here and to investigate the benefits of using more sophisticated methods, such as part of speech tagging and syntactic parsing.

Acknowledgements. I am grateful to Graeme Hirst for the advice he gave me during this work and to Melanie Baljko, Phil Edmonds, and Stephen Green for their help with the experimental work.

This research was conducted while I was at the University of Toronto, and was supported by the Natural Sciences and Engineering Research Council of Canada.

References

Jean Caron. 1997. Toward a procedural approach of the meaning of connectives. In Jean Costermans and Michel Fayol, editors, *Processing Interclausal Relationships. Studies in the Production and Comprehension of Text*, pages 53–74. Lawrence Erlbaum Associates.

Dan Cristea and Bonnie L. Webber. 1997. Expectations in incremental discourse processing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL/EACL–97)*, pages 88–95, Madrid, Spain, July 7–12.

Bruce Fraser. 1996. Pragmatic markers. *Pragmatics*, 6(2):167–190.

Barbara Grosz and Julia Hirschberg. 1992. Some intonational characteristics of discourse structure. In *Proceedings of the International Conference on Spoken Language Processing*.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, July–September.

Marti A. Hearst. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, March.

Julia Hirschberg and Diane Litman. 1993. Empirical studies on the disambiguation of cue phrases. *Computational Linguistics*, 19(3):501–530.

Diane J. Litman. 1996. Cue phrase classification using machine learning. *Journal of Artificial Intelligence Research*, 5:53–94.

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

Daniel Marcu. 1997a. From discourse structures to text summaries. In *Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 82–88, Madrid, Spain, July 11.

Daniel Marcu. 1997b. *The rhetorical parsing, summarization, and generation of natural language texts*. Ph.D. thesis, Department of Computer Science, University of Toronto, December.

Leo G.M. Noordman and Wietske Vonk. 1997. Toward a procedural approach of the meaning of connectives. In Jean Costermans and Michel Fayol, editors, *Processing Interclausal Relationships. Studies in the Production and Comprehension of Text*, pages 75–93. Lawrence Erlbaum Associates.

Rebbeca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–140, March.

Eric V. Siegel and Kathleen R. McKeown. 1994. Emergent linguistic rules from inducing decision trees: Disambiguating discourse clue words. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI–94)*, volume 1, pages 820–826, Seattle, July 31 – August 4.

Simone Teufel and Marc Moens. 1998. Sentence extraction and rhetorical classification for flexible abstracts. In *Working Notes of the AAAI–98 Spring Symposium on Intelligent Text Summarization*, Stanford, March 23–25.

Teun A. van Dijk. 1979. Pragmatic connectives. *Journal of Pragmatics*, 3:447–456.

Yaakov Yaari. 1997. Segmentation of expository texts by hierarchical agglomerative clustering. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP–97)*, Bulgaria.