Beyond String Matching and Cue Phrases: Improving Efficiency and Coverage in Discourse Analysis

Simon H. Corston-Oliver

Microsoft Research
One Microsoft Way
Redmond WA 98052-6399
USA
simonco@microsoft.com

Abstract

RASTA (Rhetorical Structure Theory Analyzer), a discourse analysis component within the Microsoft English Grammar, efficiently computes representations of the structure of written discourse using information available in syntactic and logical form analyses. RASTA heuristically scores the rhetorical relations that it hypothesizes, using those scores to guide it in producing more plausible discourse representations before less plausible ones. The heuristic scores also provide a genre-independent method for evaluating competing discourse analyses: the best discourse analyses are those constructed from the strongest hypotheses.

Introduction

Rhetorical Structure Theory (RST) (Mann and Thompson 1986, 1988) models the discourse structure of a text by means of a hierarchical tree diagram that labels relations between propositions expressed in text spans (typically clauses or larger linguistic units). The relations between nodes are of two kinds: *symmetric* and *asymmetric*. A symmetric relation involves two or more nodes, conventionally labeled *nuclei*, each of which is equally important in realizing the writer's communicative goals. An asymmetric relation involves exactly two nodes: a *nucleus*, the more important of the two in realizing the writer's communicative goals, and a *satellite*, a node in a dependency relation to the nucleus, modifying the nucleus in ways specified in the definition of the particular relation. Hierarchical structure arises from the fact that a nucleus or satellite node may have internal structure. As Mann and Thompson (1988:266-268) note, the hierarchical representations of RST are well-suited to the task of text summarization: if satellite nodes occurring below a certain depth in an RST tree were omitted, the text that remained would form a coherent synopsis of the author's main points. A reliable, efficient method for constructing RST representations is therefore a prerequisite for one approach to text summarization.

A human analyst performing an RST analysis of a text proposes a judgment of the plausibility that the writer intended a certain effect (Mann and Thompson 1988:245). The subjective, intuitive nature of such judgments and the lack of an explicit method for constructing analyses lead Sanders and van Wijk (1996:94) to conclude that "Rhetorical Structure Theory lacks a procedure." In contrast to the intuitive nature of human judgements, a computational model for constructing RST analyses requires explicit procedures for two purposes: (1) to recognize discourse relations and (2) to construct RST trees on the basis of those relations.

RASTA (Rhetorical Structure Theory Analyzer) is a component of the Microsoft English Grammar (MEG) that constructs RST analyses for texts. To date the input to RASTA has been articles in the *Encarta 96 Encyclopedia* (Microsoft Corporation 1995). Although these articles have been edited to conform to a style guide governing content and lexical choice, they exhibit a wide range of discourse structures. As is generally true of encyclopedia articles (Maier and Hovy 1991:6), *ideational* and *textual* relations (as well as a single *interpersonal* relation, CONCESSION) are employed to realize a speech act such as DESCRIBE or EXPLAIN. The analysis of *Encarta* fits within the ongoing research program of the Natural Language Processing Group at Microsoft Research to extract structured knowledge representations from reference works intended for human readers (Dolan, Vanderwende and Richardson 1993; Richardson, Vanderwende and Dolan 1993; Dolan 1995; Vanderwende 1995a, 1995b; Richardson 1997).

Recognizing Discourse Relations

The discourse literature contains a full spectrum of views on how to recognize discourse relations, ranging from unconstrained reference to world knowledge (Hobbs 1979; Polanyi 1988) or reasoning with representations of propositional content (Wu and Lytinen 1990; Fukumoto and Tsujii 1994) to a reliance on cue words or phrases and lexical repetition (Sumita et al. 1992; Kurohashi and Nagao 1994; Ono, Sumita and Miike 1994; Marcu 1997). A method for recognizing relations from a direct examination of a text is surely to be preferred over methods predicated on propositional representations, since the question of how to extract such propositional representations or model world knowledge is by no means resolved.

Superficial techniques, using regular expressions to perform string matching, encounter several problems. First, it is difficult to reliably identify textual units that ought to serve as terminal nodes in an RST analysis. For example, Marcu (1997), the most complete account to date of a technique for identifying discourse relations from a direct examination of a text, incorrectly identifies as terminal nodes such constituents as "50 percent farther from the sun than the Earth"—constituents that are not even clauses, and that would therefore not be treated as terminal nodes in a conventional RST analysis.

The second problem with pattern matching techniques, is that many cue phrases in *Encarta* are amenable to two syntactic interpretations. For example the string "as long as" ought to be treated as a single lexical item, equivalent to a subordinating conjunction, in the *Encarta* excerpt "The premier and cabinet remain in power as long as they have the support of a majority in the provincial legislature"; however, in the sentence "...their observed light would have been traveling practically as long as the age of the universe" it is a phrase with internal syntactic constituency. While the distinction between a compositional and non-compositional reading of this string falls out of the syntactic analysis performed by MEG, it would be extremely difficult to identify by means of regular-expression matching given the typographically identical environments.

Redeker (1990) finds cue phrases in approximately fifty percent of clauses, a figure which Marcu (1997:97) interprets as "sufficiently large to enable the derivation of rich rhetorical structures for texts." Viewing the proverbial glass as half empty, we might instead ask how we could identify discourse relations in the absence of cue phrases.

RASTA combines cue phrase identification with an examination of the syntactic analysis and logical form (a normalized syntactic analysis with the flavor of a predicate-argument representation) for a text to identify discourse relations. Since all the possible rhetorical relations that might hold between two nodes are not equally likely RASTA assigns heuristic scores to its judgments. RASTA tests to see whether the necessary criteria for the relation are satisfied. If they are satisfied, RASTA tests each of the optional cues that might help to identify that relation. Each cue has an associated heuristic score. The heuristic score associated with a given rhetorical relation is equal to the sum of heuristic scores for the cues that were satisfied for that relation.

Before identifying rhetorical relations, RASTA identifies all the clauses that might function as terminal nodes in an RST analysis, excluding subject and object complements and other clauses that do not have "independent functional integrity" (Mann and Thompson 1988:248). RASTA has available to it the full syntactic analysis performed by MEG. RASTA then examines pairs of clauses, labeled Clause$_1$ and Clause$_2$, to determine what rhetorical relations might hold between them. The following thirteen relations are sufficient for the analysis of *Encarta* articles. ASYMMETRICCONTRAST, CAUSE, CIRCUMSTANCE, CONCESSION, CONDITION, CONTRAST, ELABORATION, JOINT, LIST, MEANS, PURPOSE, RESULT, SEQUENCE. The identification of the ELABORATION and CAUSE relations is illustrated here.

Figure 1 gives the necessary criteria for ELABORATION, an asymmetric relation. If these necessary criteria are satisfied, RASTA tests the cues given in Figure 2. The term "Dobj" refers to a node in the logical form that represents the normalization of two syntactic roles: the direct object of a transitive construction and the subject of a passive or unaccusative construction. Subject continuity is determined either by lexical repetition or by the anaphora resolution component of MEG.

1.1 Clause$_1$ precedes Clause$_2$.
1.2 Clause$_1$ is not syntactically subordinate to Clause$_2$.
1.3 Clause$_2$ is not syntactically subordinate to Clause$_1$.

Figure 1 Necessary criteria for the ELABORATION relation

As Figure 2 shows, RASTA tends to assign high heuristic scores to cue words and phrases as indicators of a relation, and lower scores to syntactic evidence. In many cases, however, several syntactic cues will converge to assign a high heuristic score to a given relation.

2.1. Clause$_1$ is the main clause of a sentence (*sentence$_i$*) and Clause$_2$ is the main clause of a sentence (*sentence$_j$*) and *sentence$_i$* immediately precedes *sentence$_j$* and (a) Clause$_2$ contains an elaboration conjunction (*also for_example*) or (b) Clause$_2$ is in a coordinate structure whose parent contains an elaboration conjunction. Score = 35.
2.2. Cue (2.1) applies and Clause$_1$ is the main clause of the first sentence in the excerpt. Score = 15.
2.3. Clause$_2$ contains a predicate nominal whose head is in the set {*portion component member type kind example instance*} or Clause$_2$ contains a predicate whose head verb is in the set {*include consist*}. Score = 35.
2.4. Clause$_1$ and Clause$_2$ are not coordinated and (a) Clause$_1$ and Clause$_2$ exhibit subject continuity or (b) Clause$_2$ is passive and the head of the Dobj of Clause$_1$ and the head of the Dobj of Clause$_2$ are the same lemma (i.e. citation form) or (c) Clause$_2$ contains an elaboration conjunction. Score = 10.
2.5. Cue (2.4) applies and Clause$_2$ contains a habitual adverb (*sometimes usually...*). Score = 17.
2.6. Cue (2.4) applies and the syntactic subject of Clause$_2$ is the pronoun *some* or contains the modifier *some*. Score = 10.

Figure 2 Cues to the ELABORATION relation

Figure 3 gives the necessary criteria for the CAUSE relation, a symmetric relation. Figure 4 gives the cues that are tested if the necessary criteria are satisfied.

3.1. Clause$_1$ precedes Clause$_2$.
3.2. Clause$_1$ is not syntactically subordinate to Clause$_2$.
3.3. Clause$_2$ is not syntactically subordinate to Clause$_1$.
3.4. The subject of Clause$_2$ is not a demonstrative pronoun, nor is it modified by a demonstrative.

Figure 3 Necessary criteria for the CONTRAST relation

4.1. Clause$_2$ is dominated by or contains a contrast conjunction (*but however or...*). If Clause$_2$ is in a coordinate structure, then it must be coordinated with Clause$_1$. Score = 25.
4.2. Cue (4.1) is satisfied and the head verbs of Clause$_1$ and Clause$_2$ have the same lemma. Score = 10.
4.3. Clause$_1$ and Clause$_2$ differ in polarity (i.e. one clause is positive and the other negative). Score = 5.
4.4. The syntactic subject of Clause$_1$ is the pronoun *some* or has the modifier *some* and the subject of Clause$_2$ is the pronoun *other* or has the modifier *other*. Score = 30.

Figure 4 Cues to the CONTRAST relation

5.1. The aardwolf is classified as Proteles cristatus.
5.2. It is usually placed in the hyena family, Hyaenidae.
5.3. Some experts, however, place the aardwolf in a separate family, Protelidae, because of certain anatomical differences between the aardwolf and the hyena
5.4. For example, the aardwolf has five toes on its forefeet...

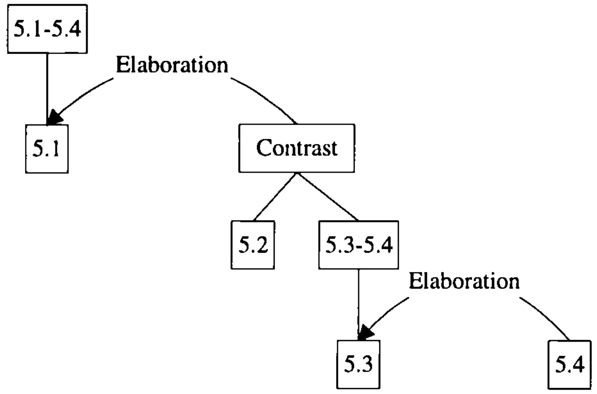

Figure 5 Aardwolf

The cues for the ELABORATION and CONTRAST relations are sufficient to lead to the construction of the plausible RST analysis given in Figure 5. Cues 2.4 and 2.5 lead RASTA to hypothesize an ELABORATION relation between clauses 5.1 and 5.2. An ELABORATION relation is also posited between clauses 5.1 and 5.3 (cues 2.4 and 2.6) and between clauses 5.3 and 5.4 (cue 2.1). A CONTRAST relation is posited between clauses 5.2 and 5.3 on the basis of cues 4.1 and 4.2. An additional CONTRAST relation is posited between clauses 5.1 and 5.3 on the basis of cue 4.1. This relation, however, is not employed in constructing the most plausible RST tree.

It is important to emphasize that RASTA merely needs to distinguish among the thirteen rhetorical relations employed. The criteria employed by RASTA are by no means an exhaustive analysis of the linguistic correlates of each relation.

Efficiently Constructing RST Trees

Marcu (1996) provides a first-order formalization of RST trees, with an algorithm for constructing all the RST trees compatible with a set of hypothesized rhetorical relations for a text. Marcu employs the notion of nuclearity in developing this algorithm, observing that two adjacent text spans can be related by an RST relation if and only if that relation holds between the nuclei of the two text spans; satellites of the text spans do not enter into the determination of this relationship. RST trees can thus be assembled from the bottom up by joining text spans whose nuclei have been posited to be potentially in some rhetorical relationship. Given a set of rhetorical relations that might hold between pairs of RST terminal nodes, Marcu's algorithm will attempt to apply the relations in all possible permutations to produce all of the valid RST trees that are compatible with the relations posited.

Marcu introduces the notion of a *promotion set*. The promotion set of a terminal node is the node itself. The promotion set of an asymmetric relation is equal to the promotion set of the nucleus. The promotion set of a symmetric relation is equal to the union of promotion sets of the conuclei. In Figure 5, for example, the promotion set of the text span 5.3-5.4 is the node 5.3, the promotion set of the CONTRAST relation is nodes 5.2 and 5.3, and the promotion set of the entire text is node 5.1. Marcu formalizes the following constraint on RST trees: two nodes, *a* and *b*, can be combined in a given rhetorical relation if and only if that relation holds between all members of the promotion set of *a* and all members of the promotion set of *b*. In Figure 5, for example, the text span 5.2-5.4, whose promotion set contains nodes 5.2 and 5.3, can only function as a satellite in an ELABORATION relation to node 5.1 if it is plausible to posit an ELABORATION relation with 5.1 as the nucleus and 5.2 as the satellite and another ELABORATION relation with 5.1 as the nucleus and 5.3 as the satellite. Since RASTA did posit these two relations, it is reasonable to make the CONTRAST node a satellite of 5.1 in an ELABORATION relation.

Marcu's algorithm represents a major advance in the automated construction of RST analyses on the basis of a set of hypothesized relations. However, the algorithm suffers from combinatorial explosion—as the number of hypothesized relations increases, the number of possible RST trees increases exponentially. This problem is exacerbated by the fact that Marcu first produces all possible combinations of nodes according to the relations posited, and only then culls ill-formed trees.

To overcome these problems RASTA implements a backtracking algorithm that applies the relations with the highest heuristic scores first in the bottom-up construction of RST trees.

During the bottom up construction of an RST tree, the application of an RST relation to join two nodes precludes the application of any other RST relation to join those same two nodes. RASTA therefore gathers mutually exclusive relations into *bags*. All of the relations in a bag specify rhetorical relations between the same two terminal nodes. The relations within a bag are sorted in descending order according to heuristic score. RASTA then constructs a list of bags, ALLBAGS, sorted in descending order according to the heuristic score of their first element. RASTA also maintains a simple unordered list, ALLHYPOTHS, containing all of the relations posited. Having considered all pairs of terminal nodes and posited rhetorical relations, RASTA constructs the possible RST trees by calling the procedure CONSTRUCTTREE (Figure 6) with two arguments: the list ALLBAGS as the HYPOTHESES argument and the list of terminal nodes as the SUBTREES arguments. The function Count returns the number of elements in a list. The data structure used in constructing RST representations contains, among other fields, a field called *Pod*, containing the heuristic score for the relation between the constituent nodes, and a field called *Value*, containing a heuristic score for the entire subtree.

CONSTRUCTTREE builds on the algorithm presented in Marcu (1996), but is considerably more efficient. When the algorithm detects that joining two text spans would lead to an ill-formed tree, it backtracks, thus avoiding the construction of a great many subsequent ill-formed trees. In attempting to apply relations with higher heuristic scores before relations with lower scores, CONSTRUCTTREE tends to converge on more plausible analyses first. In principle, the procedure can be left to run until it has constructed all possible RST trees compatible with the set of hypothesized relations. In practice, however, execution can be stopped after a handful of trees has been produced, since the most plausible analyses tend to be produced early on.

Occasionally, hypotheses are not applied in strict order of their heuristic scores. Consider, for example, the following three bags of hypotheses:

1. Bag A, containing a_1, score = 15, a_2, score = 7 and a_3, score = 2
2. Bag B, containing b_1, score 10 and b_2, score 5
3. Bag C, containing c_1, score 5 and c_2, score 3.

Taking one relation from each bag yields the sequences $\{a_1\ b_1\ c_1\}$, $\{a_1\ b_1\ c_2\}$, $\{a_1\ b_2\ c_1\}$ and $\{a_1\ b_2\ c_2\}$. The next sequence $\{a_2\ b_1\ c_1\}$, however, is not in strict descending order, since a_2 has a heuristic score of 7 and b_1 has a heuristic score of 10. Rather than overburden the algorithm with an elaborate procedure for ensuring that the hypotheses are always tried in the correct order, CONSTRUCTTREES tolerates this occasional deviation, and compensates by sorting the final output according to the Value attribute of the root node of each tree.

CONSTRUCTTREE constructs binary-branching RST trees. After the completion of CONSTRUCTTREE, RASTA performs a top-down traversal of the trees, transforming them from binary-branching representations to the more standard n-ary branching representations.

```
Function ConstructTree (HYPOTHESES, SUBTREES)
Begin Function ConstructTree

Let COPYHYPOTHS be a copy of the list HYPOTHESES.

If Count(SUBTREES) == 1 Then
  Store this RST tree if it is not identical to one already stored.
  Return.
Else If Count(COPYHYPOTHS) >= 1 and Count(SUBTREES) > 1
Then
  For each bag, ONEBAG, in COPYHYPOTHS
    Let BAGSLEFT be a copy of COPYHYPOTHS except ONEBAG.

    If promotions of elements in SUBTREES contain the nodes
      specified by the relations in ONEBAG Then
      For each hypothesis, ONEHYP, in ONEBAG
        1.  Let NUC be the subtree whose promotion set
            includes the leftmost nucleus specified by
            ONEHYP.
        2.  Let OTHER be the subtree whose promotion set
            includes the other node specified by ONEHYP.
        3.  Search ALLHYPOTHESES. The relation specified by
            ONEHYP must have been hypothesized between
            every member of the promotion set of NUC and
            every member of the promotion set of OTHER.
        If (3) is true
          If combining NUC and OTHER yields an ill-formed
          tree Then return.
          Else
            Let SUBTREESLEFT be a copy of SUBTREES, with
              NUC and OTHER removed.
            Create a new subtree, NEWTREE, by joining NUC
              and OTHER as specified by ONEHYP.
            Set Pod(NEWTREE ) equal to the heuristic score
              of ONEHYP.
            Set Value(NEWTREE) equal to Pod(NEWTREE) +
              Value(NUC) + Value(OTHER).
            Prepend NEWTREE to REMAININGSUBTREES.
            ConstructTree (BAGSLEFT, SUBTREESLEFT).
          End If.
        End If
      Do the next element in ONEBAG until all elements done.
    Else // this bag cannot apply in any subsequent permutation
      Remove ONEBAG from COPYHYPOTHS
    End If
  Do the next bag until Count(COPYHYPOTHS) == 0
Else // HYPOTHESES is empty
   Return.
End If

End Function ConstructTree
```

Figure 6 Pseudo-code for constructing RST trees

Evaluating Alternative Analyses

Marcu (1997) claims that right-branching structures ought to be preferred because they reflect basic organizational properties of text. In fact, the success of this metric reflects the genre of his three test excerpts. Two of the test excerpts are from magazines, which are widely known to have a concatenative structure, as Marcu (1997:100) himself observes. The third text is a brief narrative, whose right-branching structure is perhaps a reflection of iconic principles of organization (Haiman 1980). In a narrative, the linear order of clauses matches the temporal sequence of events (Labov 1972). Narratives can thus be said to unfold in a right-branching manner.

RASTA sorts the output of CONSTRUCTTREE according to the Value attribute of the root node of each tree, the premise being that the best RST trees are those constructed from relations with the highest heuristic scores. Thus, RASTA uses the heuristic scores for two purposes: (1) to guide it in generating trees, and (2) to evaluate and rank the trees produced.

Learning Heuristic Scores

The heuristic scores assigned to individual cues have been developed by trial and modification, with the initial values based on the author's intuitions as a linguist. For example, conjunctions are extremely good discriminators of particular discourse relations, whereas tense and aspect are weaker discriminators. As new data have been encountered, these values have been modified to ensure that preferred analyses occur at the top of the ranked list of RST trees.

Researchers developing grammars have access to annotated corpora such as the Penn Treebank (Marcus, Santorini and Marcinkiewicz 1993) that can be used for the automated learning of patterns or to evaluate grammars. Unfortunately, no corpora annotated for RST relations currently exist. Hand-tuning in order to determine the optimal heuristic scores is therefore still necessary. Since the researcher is able to draw on linguistic intuitions to constrain the space that must be searched for ideal scores, this task is not as daunting as it might appear. Furthermore, preliminary statistical analysis of the performance of RASTA suggests that machine learning would not substantially improve the performance of RASTA. Because great care has been taken in developing the necessary criteria for identifying rhetorical relations, RASTA tends not to produce spurious hypotheses and therefore often produces a single RST analysis for a text. Preliminary measurements of one test set consisting of 59 excerpts from *Encarta* have shown that on average, the preferred analysis was produced within 1.92 trees, with a standard deviation of 4.39 (Corston-Oliver 1998). Therefore, no more than a dozen trees need to be produced to ensure that the best tree will have been produced. In all cases, the preferred tree percolated to the top when the trees were sorted according to the Value attribute.

Extending RASTA

RASTA consists of two phases. During the first phase, RASTA posits discourse relations between terminal nodes. This phase could easily be extended by adding cues to identify discourse relations in another genre. For example, for processing office memos cues could be developed to identify numbered list structures.

During the second phase, RASTA constructs RST trees compatible with the relations posited during the first phase. Nothing in the second phase is dependent on particular relations. For example, a slightly different set of RST relations could be employed (perhaps distinguishing VOLITIONAL CAUSE and NON-VOLITIONAL CAUSE, as per Mann and Thompson 1986), without requiring any changes in the second phase. Similarly, RASTA's evaluation metric, which favors the RST trees constructed from the relations with the highest heuristic scores, is intended to apply to any genre, although it may prove to be the case that the exact heuristic scores associated with each cue might require modification in different genres.

One well-known problem for RST is that a text can simultaneously have both an intentional and an informational representation (Ford 1986; Moore and Pollack 1992), but that these intentional and informational representations will not necessarily have the same structure. RST is not able to represent this possible mismatch between intentional and informational representations, since it requires that an analyst choose one relation to relate two text spans, necessitating a choice between *either* an intentional relation *or* an informational relation. The heuristic scores associated with relations in RASTA mitigate this problem: both an intentional and an informational relation can be posited, each with its own heuristic score. The heuristic scores could even be varied according the goal of the analysis. For example, if the discourse analysis was intended as input for text summarization, it might prove desirable to favor informational relations.

Conclusion

The development of RASTA has been guided by a functionalist approach to analyzing language. Writers employ linguistic resources—morphology, the lexicon, syntax—to realize their communicative goals. In employing these linguistic resources, a text is molded, taking on a specific form from which it is possible to infer the writer's communicative goals. Human readers, no doubt, employ knowledge outside of a text to aid in its interpretation, drawing on such factors as world knowledge, genre conventions and plausible inferences. Rather than attempting to model such extrinsic knowledge and thereby mimic the current understanding of the mental processes of human readers, RASTA proceeds under the assumption that the text itself contains sufficient cues to enable a computer to construct a feasible representation of its discourse structure. The kinds of cues employed by RASTA for the analysis of *Encarta* are expected to apply equally well to other genres, although additional cues might also be required. Finally, although reasoning beyond linguistic form might prove necessary at some later point, the computationally (and philosophically) more straightforward approach to identifying formal cues outlined here appears to hold considerable promise.

Acknowledgments. I wish to thank Mike Barnett, George Heidorn and Miguel de Campos for helpful discussion and programming assistance in improving the efficiency of RASTA. My thanks go also to Monica Corston-Oliver, Robert Stumberger and the members of the Natural Language Processing Group at Microsoft Research for comments on an earlier draft of this paper.

References

Corston-Oliver, S. H. 1998. Computing Representations of Discourse Structure. Ph.D. diss., Dept. of Linguistics, University of California, Santa Barbara. Forthcoming.

Dolan, W. B. 1995. Metaphor as an emergent property of machine-readable dictionaries. In Proceedings of the AAAI 1995 Spring Symposium Series: Representation and Acquisition of Lexical Knowledge: Polysemy, Ambiguity and Generativity, 27-32.

Dolan, W. B., Vanderwende, L. and Richardson, S. D. 1993. Automatically deriving structured knowledge bases from on-line dictionaries. In Proceedings of the Pacific Association for Computational Linguistics (PACLING '93), Vancouver, British Columbia, 5-14.

Ford, Cécilia E. 1986. Overlapping relations in text structure. In DeLancey, Scott and Russell S. Tomlin (eds.), *Proceedings of the Second Annual Meeting of the Pacific Linguistics Conference*. 107-123.

Fukumoto, J. and Tsujii, J. 1994. Breaking down rhetorical relations for the purpose of analyzing discourse structures. In COLING 94: The Proceedings of the 15th International Conference on Computational Linguistics, vol. 2:1177-1183.

Haiman, J. 1980. The Iconicity of Grammar: Isomorphism and Motivation. *Language* 56:515-540.

Hobbs, J. R. 1979. Coherence and coreference. *Cognitive Science* 3:67-90.

Kurohashi, S. and Nagao, M. 1994. Automatic detection of discourse structure by checking surface information in sentences. In *Proceedings of COLING 94: The 15th International Conference on Computational Linguistics*, vol. 2:1123-1127.

Labov, W. 1972. *Language in the Inner City: Studies in the Black English Vernacular—Conduct and Communication.* Philadelphia: University of Pennsylvania Press.

Maier, E. and Hovy, E. H. 1991. A metafunctionally motivated taxonomy for discourse structure relations. Ms.

Mann, W. C. and Thompson, S. A. 1986. Relational Propositions in Discourse. *Discourse Processes* 9:57-90.

Mann, W. C. and Thompson, S. A. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text* 8:243-281.

Marcu, D. 1996. Building Up Rhetorical Structure Trees. In Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), volume 2:1069-1074.

Marcu, D. 1997. The Rhetorical Parsing of Natural Language Texts. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL/EACL-97), 96-103.

Marcus, M. P., Santorini B. and Marcinkiewicz, M. A. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19:313-330.

Microsoft Corporation. 1995. Encarta® 96 Encyclopedia. Redmond: Microsoft.

Moore, Johanna D. and Martha E. Pollack. 1992. A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics* 18:537-544

Ono, K., Sumita, K. and Miike, S. 1994. Abstract generation based on rhetorical structure extraction. In Proceedings of COLING 94: The 15th International Conference on Computational Linguistics, vol. 2:344-348.

Polanyi, L. 1988. A formal model of the structure of discourse. *Journal of Pragmatics* 12:601-638.

Redeker, G. 1990. Ideational and pragmatic markers of discourse structure. *Journal of Pragmatics* 14:367-381.

Richardson, S. D. 1997. Determining Similarity and Inferring Relations in a Lexical Knowledge Base. Ph.D. diss., Dept. of Computer Science, The City University of New York.

Richardson, S. D., Vanderwende, L. and Dolan, W. 1993. Combining dictionary-based methods for natural language analysis. In Proceedings of the TMI-93, Kyoto, Japan, 69-79.

Sanders, T. J. M. and Wijk, C. van. 1996. PISA: A procedure for analyzing the structure of explanatory texts. *Text* 16:91-132.

Sumita, K., Ono, K., Chino, T., Ukita, T. and Amano, S.

1992. A discourse structure analyzer for Japanese text. In Proceedings of the International Conference of Fifth Generation Computer Systems, 1133-1140.

Vanderwende, L. H. 1995a. The Analysis of Noun Sequences Using Semantic Information Extracted from On-Line Dictionaries. Ph.D. diss., Dept. of Linguistics, Georgetown University.

Vanderwende, L. H. 1995b. Ambiguity in the acquisition of lexical information. In Proceedings of the AAAI 1995 Spring Symposium Series, working notes of the symposium on representation and acquisition of lexical knowledge, 174-179.

Wu, H. J. P. and Lytinen, S. L. 1990. Coherence relatio reasoning in persuasive discourse. In Proceedings of the Twelfth Annual Conference of the Cognitive Science Society, 503-510.