# Machine-Assisted Rhetorical Structure Annotation

**Manfred Stede and Silvan Heintze**
University of Potsdam
Dept. of Linguistics
Applied Computational Linguistics
D-14415 Potsdam
Germany
stede|heintze@ling.uni-potsdam.de

## Abstract

Manually annotating the rhetorical structure of texts is very labour-intensive. At the same time, high-quality automatic analysis is currently out of reach. We thus propose to split the manual annotation in two phases: the simpler marking of lexical connectives and their relations, and the more difficult decisions on overall tree structure. To this end, we developed an environment of two analysis tools and XML-based declarative resources. Our ConAno tool allows for efficient, interactive annotation of connectives, scopes and relations. This intermediate result is exported to O'Donnell's 'RST Tool', which facilitates completing the tree structure.

## 1 Introduction

A number of approaches tackling the difficult problem of automatic discourse parsing have been proposed in recent years (e.g., (Sumita et al., 1992) (Marcu, 1997), (Schilder, 2002)). They differ in their orientation toward symbolic or statistical information, but they all — quite naturally — share the assumption that the lexical *connectives* or *discourse markers* are the primary source of information for constructing a rhetorical tree automatically. The density of discourse markers in a text depends on its genre (e.g., commentaries tend to have more than narratives), but in general, it is clear that only a portion of the relations holding in a text is lexically signalled.[1] Furthermore, it is well-known that discourse markers are often ambiguous; for example, the English *but* can, in terms of (Mann, Thompson, 1988), signal any of the relations Antithesis, Contrast, and Concession. Accordingly, automatic discourse parsing focusing on connectives is bound to have its limitations.

[1] In our corpus of newspaper commentaries (Stede, 2004), we found that 35% of the coherence relations are signalled by a connective.

Our position is that progress in discourse parsing relies on the one hand on a more thorough understanding of the underlying issues, and on the other hand on the availability of human-annotated corpora, which can serve as a resource for in-depth studies of discourse-structural phenomena, and also for training statistical analysis programs. Two examples of such corpora are the RST Tree Corpus by (Marcu et al., 1999) for English and the Potsdam Commentary Corpus (Stede, 2004) for German. Producing such resources is a labour-intensive task that requires time, trained annotators, and clearly specified guidelines on what relation to choose under which circumstances. Nonetheless, rhetorical analysis remains to be in part a rather subjective process (see section 2). In order to eventually arrive at more objective, comparable results, our proposal is to split the annotation process into two parts:

1. Annotation of connectives, their scopes (the two related textual units), and — optionally — the signalled relation
2. Annotation of the remaining (unsignalled) relations between larger segments

Step 1 is inspired by work done for English in the Penn Discourse TreeBank[2] (Miltsakaki et al., 2004). In our two-step scenario, it is the easier part of the whole task in that connectives can be quite clearly identified, their scopes are often (but not always, see below) transparent, and the coherence relation is often clear. We see the result of step 1 as a corpus resource in its own right (it can be used for training statistical classifiers, for instance) and at the same time as the input for step 2, which "fills the gaps": now annotators have to decide how the set of small trees produced in step 1 is best arranged in one complete tree, which involves assigning

[2] http://www.cis.upenn.edu/∼pdtb/

relations to instances without any lexical signals and also making more complicated scope judgements across larger spans of text — the more subjective and also more time-consuming step.[3]

Our approach is as follows. To speed up the annotation process in step 1, we have developed an XML format and a dedicated analysis tool called ConAno, which will be introduced in Section 4. ConAno can export the annotated text in the 'rs3' format that serves as input to O'Donnell's *RST Tool* (O'Donnell, 1997). His original idea was that manual annotation be done completely with his tool; we opted however to use it only for step 2, and will motivate the reasons for this overall architecture in Section 5.

The net result is a modular, XML-based annotation environment for *machine-assisted rhetorical analysis*, which we see as on the one hand less ambitious than fully-automatic discourse parsing and on the other hand as more efficient than completely 'manual' analysis.

## 2 Approaches to rhetorical analysis

There are two different perspectives on the task of discourse parsing: an "ideal" one that aims at modelling a systematic, incremental process; and an "empirical" one that takes the experiences of human annotators into account. "Ideally", discourse analysis proceeds incrementally from left to right, where for each new segment, an attachment point and a relation (or more than one of each, cf. SDRT) are computed and the discourse structure grows step by step. This view is taken for instance in SDRT (Asher, Lascarides, 2003), which places emphasis on the notion of 'right frontier' (also discussed recently by (Webber et al., 2003)).

However, when we trained two (experienced) students to annotate the 171 newspaper commentaries of the Potsdam Commentary Corpus (Stede, 2004) and upon completion of the task asked them about their experiences, a very different picture emerged. Both annotators agreed that a strict left-to-right approach is highly impractical, because the intended argumentative structure of the text often becomes clear only in retrospect, after reflecting the possible contributions of the segments to the larger scheme. Thus they very soon settled on a bottom-up approach: First, mark the transparent cases, in which a connective undoubtedly signals a relation between two segments.[4] Then, see how the resulting pieces fit together into a structure that mirrors the argument presented.

The annotators used *RST Tool* (O'Donnell, 1997), which worked reasonably well for the purpose. However, since we also have in our group an XML-based lexicon of German connectives at our disposal (Berger et al., 2002), why not use this resource to speed up the first phase of the annotation?

## 3 Annotating connectives and their scopes

In our definition of 'connective', we largely follow (Pasch et al., 2003) (a substantial catalogue and classification of German connectives), who require them to take two arguments that can potentially be full clauses and that semantically denote two-place relations between eventualities (but they need not always be spelled out as clauses). From the syntactic viewpoint, they are a rather inhomogeneous group consisting of subordinating and coordinating conjunctions, some prepositions, and a number of sententence adverbials. We refer to the two related units as an 'internal' and an 'external' one, where the 'internal' one is the unit of which the connective is actually a part. For example, in *Despite the heavy rain we had a great time*, the noun phrase *the heavy rain* is the internal unit, since it forms a syntactic phrase together with the preposition. Notice that this is a case where the eventuality (a state of weather) is not made explicit by a verb.

As indicated, this step of annotating connectives and units is closely related to the idea of the PDTB project, which seeks to develop a large corpus annotated with information on discourse structure for English texts. For this purpose, annotators are provided with detailed annotation guidelines, which point out various challenges in the annotation process for explicit as well as empty connectives and their respective arguments. They include, among others,

- words/phrases that look like connectives, but prove not to take two propositional arguments

[3] This assessment of relative difficulty does not carry over to PDTB, where the annotations are more complex than in our step 1 but do not go as far as building rhetorical structures.

[4] The clearest cases are subjunctors, which always mark a relation between matrix clause and embedded clause.

- words/phrases as preposed predicate complements
- pre- and post-modified connectives
- co-occurring connectives
- single and multiple clauses/sentences as arguments of connectives
- annotation of discontinuous connective arguments

Annotators have to also make syntactic judgements, which is not the case in our approach (where syntax would be done on a different annotation layer, see (Stede, 2004)).

In the following, we briefly explain the most important problematic issues with annotating German connectives and the way we deal with them, using our annotation scheme for ConAno.

### 3.1 Issues with German connectives

**Connective or not:** Some words can be used as connective or in other functions, such as *und* ('and'), which can for example conjoin clauses (connective) or items in a list (no connective).

**Which relation:** Some connectives can signal more than one relation, as the abovementioned *but* and its German counterpart *aber*.

**Complex connectives:** Connectives can be phrasal (e.g., *aus diesem Grund*, 'for this reason') or even discontinuous (e.g., *entweder ... oder*, 'either ... or'). A fortiori, some may be used in more than one order (*wenn* A, *dann* B / *dann* B, *wenn* A / *dann, wenn* A, B; 'if ... then ...').

**Multiple connectives/relations:** Some connectives can be joined to form a complex one, which might then signal more than one relation (e.g., combinations with *und* and *aber*, such as *aber dennoch*, 'but still').

**Modified connectives:** Some but not all connectives are subject to modification (e.g., *nur dann, wenn*, 'only then, if'; *besonders weil*, 'especially because').

**Embedded segments:** The minimal units linked by the connective may be embedded rather than adjacent: *Wir müssen, weil die Zeit drängt, uns Montag treffen* ('We have to, because time is short, meet on Monday').

### 3.2 A DTD and an Example

As the first step toward an annotation tool, we defined an XML format for texts with connectives and their scopes. Figure 1 shows the DTD, and Figure 2 a short sample annotation of a single — yet complex — sentence: *Auch Berlin koennte, jedenfalls dann, wenn der Bund sich erkenntlich zeigt, um die Notlage seiner Hauptstadt zu lindern, davon profitieren.* ('Berlin, too, could – at least if the federation shows some gratitude in order to alleviate the emergency of its capital – profit from it.') The DTD introduces XML tags for each of the connectives (`<connective>`), their possible modifiers (`<modifier>`) and respective discourse units (`<unit>`, where the `type` can be filled by `int` or `ext`), as well as the entire text (`<discourse>`). Henceforth, we will refer to the text unit containing the connective as the internal, 'int-unit' and to the other, external, one as 'ext-unit'. Using this DTD, it is possible to represent the range of problematic phenomena discussed in the previous section.

**Connective or not:** Only those words actually used as connectives will be marked with the `<connective>` tag, while others such as the frequently occurring *und* ('and') or *oder* ('or') will remain unmarked, if they merely conjoin items in a list.

**Which relation:** The `<connective>` tag includes a `rel` attribute for optional specification of the rhetorical relation that holds between the connected clauses.

**Complex connectives:** Using an XML based annotation scheme, we can easily mark phrasal connectives such as *aus diesem Grund* ('for this reason') using the `<connective>` tag. In order for discontinuous connectives to be annotated correctly, we introduce an `id` attribute that provides every connective with a distinct reference number. This way connectives such as *entweder ... oder*, ('either ... or') can be represented as belonging together. (see `<connective id="4" rel="condition">` tags in Figure 2)

**Multiple connectives/relations:** In our annotation scheme, complex connectives such as *aber dennoch*, ('but still') are treated as two distinct connectives that indicate different relations holding between the same units.

**Modified connectives:** Connective modifiers are marked with a special `<modifier>` tag, which is embedded inside the `<connective>` tag, as shown with *jedenfalls* modifying *dann* in

our example. Hence an additional `id` attribute for this tag is not necessary.

**Embedded segments:** Discourse units are marked using the `<unit>` tag, which also provides an `id` attribute. On the one hand, this is used for assigning discourse units to their respective connectives, on the other hand it provides a way of dealing with discountinuous discourse units, as the example shows.

```
<?xml version='1.0' encoding='UTF-8'?>

<!ELEMENT modifier (#PCDATA)>

<!ELEMENT connective (#PCDATA|modifier)>
<!ATTLIST connective
    id  CDATA #IMPLIED
    rel CDATA #IMPLIED>

<!ELEMENT unit
   (#PCDATA|connective|unit)*>
<!ATTLIST unit
    id   CDATA #IMPLIED
    type CDATA #IMPLIED>

<!ELEMENT discourse (#PCDATA|unit)*>
```

Figure 1: The DTD for texts-with-connectives

```
<?xml version="1.0"?>
<!DOCTYPE discourse SYSTEM "discourse.dtd">
<discourse>
  <unit type="ext" id="4">
     Auch Berlin koennte,
     <connective id="4"
      relation="condition">
        <modifier>jedenfalls</modifier>
        dann
     </connective>
     ,
  </unit>
  <unit type="int" id="4">
     <connective id="4"
      relation="condition">
        wenn
     </connective>
     der Bund sich erkenntlich zeigt, um
     die Notlage seiner Hauptstadt zu
     lindern,
  </unit>
  <unit type="ext" id="4">
     davon profitieren.
  </unit>
</discourse>
```

Figure 2: Sample annotation in XML-format

## 4 The ConAno annotation tool

A range of relatively generic linguistic annotation tools are available today, but none of them turned out suitable for our purposes: We seek a very easy-to-use, platform-independent tool for mouse-marking ranges of text and having them associated with one another. Consequently, we decided to implement our own Java-based tool, ConAno, which is geared especially to connective/scope annotation and thus can have a very intuitive user interface.

Just like discourse parsers do, ConAno exploits the fact that connectives are the most reliable source of information. Rather than attempting an automatic disambiguation, however, ConAno merely makes suggestions to the human analyst, which she might follow or discard. In particular, ConAno loads a list of (potential) connectives, and when annotation of a text begins, highlights each candidate so that the user can either confirm (by marking its scope) or discard (by mouse-click) it if it not used as a connective. Furthermore, the connective list optionally may contain associated coherence relations, which are then also offered to the user for selection. This annotation phase is thus purely data-driven: Attention is paid only to the connectives and their specific relation candidates.

To elaborate a little, the annotation process proceeds as follows. The text is loaded into the annotation window, and the first potential connective is automatically highlighted. Potential preposed or postposed modifiers, if any, of the connective are also highlighted (in a different color). The user moves with the mouse from one connective to the next and

- can with a mouseclick discard a highlighted item (it is not a connective or not a modifier),
- can call up a help window explaining the syntactic behavior and the relations of this connective,
- can call up a suggestion for the int-unit (i.e., text portion is highlighted),
- can analogously call up a suggestion for the ext-unit,
- can choose from a menu of the relations associated with this connective.

A screenshot is given in Figure 4. The suggestions for int-unit and ext-unit are made by ConAno on the basis of the syntactic category of

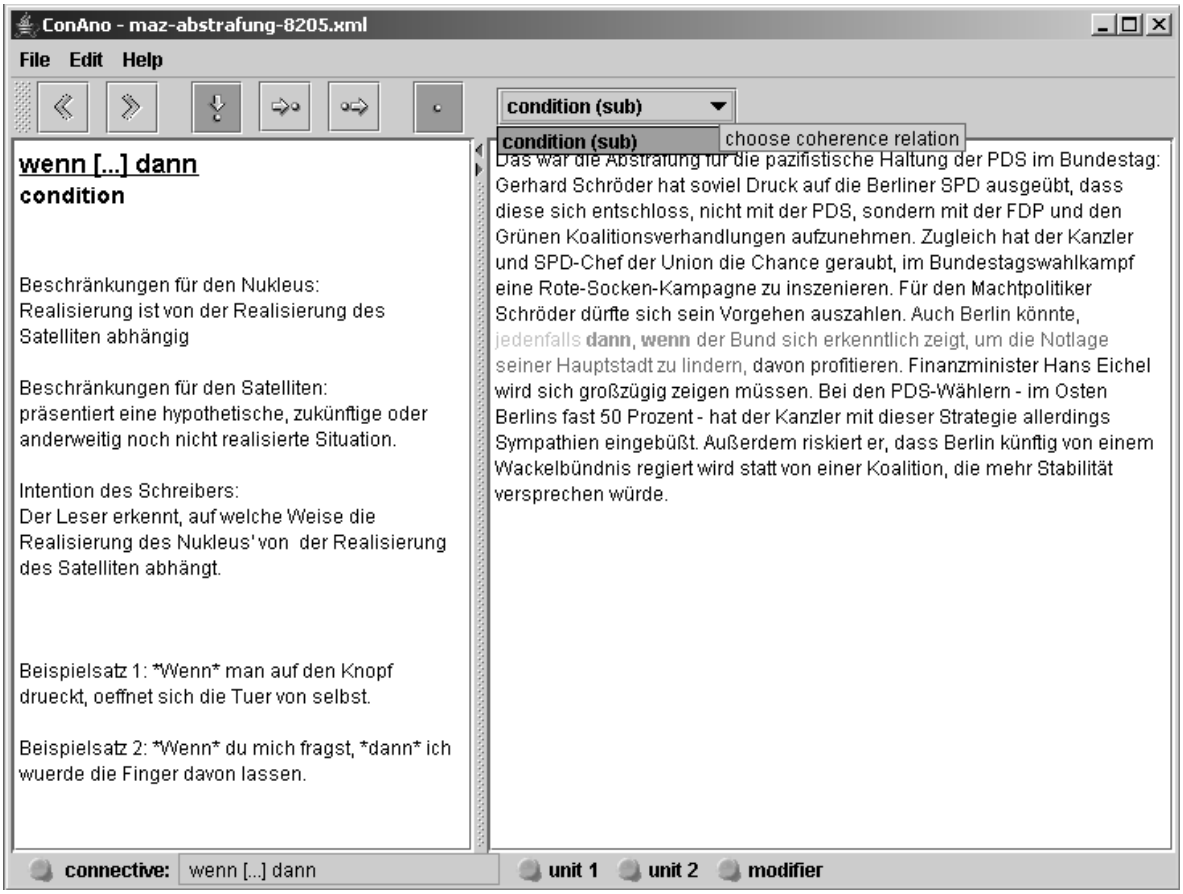


Figure 3: Screenshot of ConAno

the connective; we use simple rules like "search up to the next comma" to find the likely int-unit for a subjunctor, or "search the preceding two full-stops" to find the ext-unit for an adverbial (the preceding sentence). The suggestions may be wrong; then the user discards them and marks them with the mouse herself. The result of this annotation phase is an XML file like the (very short) one shown in Figure 2.

## 5 Overall annotation environment

A central design objective is to keep the environment neutral with respect to the languages of the text, the connectives to be annotated, and the coherence relations associated with them. Accordingly, the list of connectives is external and read into ConAno upon startup. In our case, we use an XSLT sheet to map our 'Discourse Marker Lexicon' (see below) to the input format of ConAno. The text to be annotated is expected in plain ASCII. When annotation is complete, the result (or an intermediate result) can be saved in our XML-format introduced in section 3.2. Optionally, it can be exported to the 'rs3' format developed by (O'Donnell, 1997) for his *RSTTool*. This allows for a smooth tran-

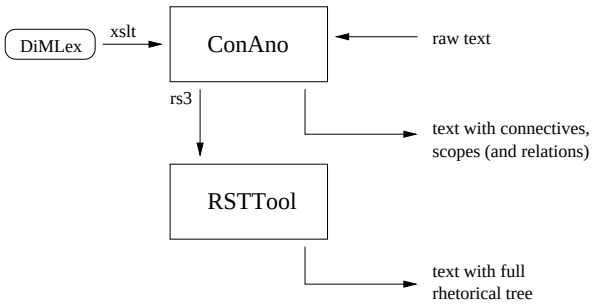


Figure 4: Overview of annotation environment

sition to a tool for constructing complete rhetorical trees. Rather than starting from scratch, the *RSTTool* user can now open the file produced by ConAno, which amounts to a partial rhetorical analysis of the text, and which the user can now complete to a full tree.

Our Discourse Marker Lexikon 'DiMLex' (Berger et al., 2002) assembles information on 140 German connectives, giving a range of syntactic, semantic, and pragmatic features, including the coherence relations along the lines of (Mann, Thompson, 1988). They are encoded in an application-neutral XML format (see Figure 5), which are mapped with XSLT sheets to various NLP applications. Our new proposal here is to use it also for interactive connective annotation. Hence, we wrote an XSLT sheet that maps DiMLex to a reduced list, where each connective is associated with syntactic labels *coordination, subordination* or *adverb* and `<coh-relation>` entries for its potential relations — see Figure 6 for DTD and 7 for an example. The, for these purposes quite simple, `syn` value has been mapped from the more complex classification in DiMLex under `kat` (German for *category*). This format is the input to ConAno.

As indicated above, we do not see the transition to *RSTTool* as a necessary step. Rather, the intermediate result of connective/scope annotation is useful in its own right, as it encodes those aspects of rhetorical structure that are independent of the chosen set of coherence relations and the conditions of assigning them.

## 6 Summary

With our work on German discourse connectives, the structure of their argument units, and the indicated rhetorical relations, we seek a better understanding of underlying linguistic issues on the one hand, and an easier way of developing rhetorical structure-annotated corpora for German texts on the other hand. For

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl"
  href="short_dictionary.xsl" ?>
<!DOCTYPE dictionary SYSTEM "dimlex.dtd">
<dictionary>
   <entry id="41">
      <orth phrasal="0">denn</orth>
      <syn>
         <kat>konj</kat>
         <position>vorvorfeld</position>
         <!-- . . . -->
      </syn>
      <semprag>
         <relation>cause</relation>
         <relation>explanation</relation>
         <presupp>int-unit</presupp>
         <!-- . . . -->
      </semprag>
      <example>
         Das Konzert muss ausfallen,
         *denn* die Saengerin ist erkrankt.
      </example>
      <example>
         Die Blumen auf dem Balkon sind
         erfroren, *denn* es hat heute
         nacht Frost gegeben.
      </example>
   </entry>
</dictionary>
```

Figure 5: DiMLex extract

this purpose, we present an annotation environment, including our ConAno tool, which helps human annotators to mark discourse connectives and their argument units by finding possible connectives and making suggestions on their estimated argument structure. We pointed out several challenges in the connective annotation process of German texts and introduced an XML based annotation scheme to handle the difficulties. For one thing, the results of this step provide elobarate information about the structure of German texts with respect to discourse connectives, but furthermore they can be used as input to O'Donnell's *RST Tool*, in order to complete the annotation of the rhetorical tree structure. The overall scenario is then one of *machine-assisted rhetorical structure annotation*. Since ConAno is based on an external list of connectives (with associated syntactic labels and relations), the tool is not dedicated to one particular theory of discourse structure, let alone to a specific set of relations. Furthermore, it can in principle deal with texts in various languages (it just relies on string matching between

```
<?xml version='1.0' encoding='UTF-8'?>

<!ELEMENT example (#PCDATA)>

<!ELEMENT coh-relation (#PCDATA)>

<!ELEMENT sem (example|coh-relation)*>

<!ELEMENT syn (sem)*>
<!ATTLIST syn
    type CDATA #IMPLIED>

<!ELEMENT part (#PCDATA)>
<!ATTLIST part
    type CDATA #IMPLIED>

<!ELEMENT orth (part)*>
<!ATTLIST orth
    type CDATA #IMPLIED>

<!ELEMENT entry (syn|orth)*>
<!ATTLIST entry
    id CDATA #IMPLIED>

<!ELEMENT conanolex (entry)*>
```

Figure 6: DTD for connectives in ConAno input format

```
<entry id="116">
   <orth type="cont">
      <part type="single">wenn</part>
   </orth>
   <orth type="discont">
      <part type="single">wenn</part>
      <part type="single">dann</part>
   </orth>
   <syn type="subordination">
      <sem>
         <coh-relation>condition
         </coh-relation>
         <example>*Wenn* man auf den Knopf
           drueckt, oeffnet sich die Tuer
           von selbst.
         </example>
         <example>*Wenn* du mich fragst,
           *dann* wuerde ich die Finger
           davon lassen.
         </example>
      </sem>
   </syn>
</entry>
```

Figure 7: Connective information in ConAno input format

connectives in the list and in the text), but we have so far used it only for German.


## Acknowledgements

We thank the anonymous reviewers for their constructive comments and suggestions for improving the paper.


## References


Asher, N. and Lascarides, A. 2003. *Logics of Conversation.* Cambridge University Press.

Berger, D.; Reitter, D. and Stede, M. 2002. XML/XSL in the Dictionary: The Case of Discourse Markers. In: Proc. of the Coling Workshop 'NLPXML-2002', Tapei.

O'Donnell, M. 1997. RST-Tool: An RST Analysis Tool. Proc. of the 6th European Workshop on Natural Language Generation, Duisburg.

Mann, W. and Thompson, S. 1988. Rhetorical Structure Theory: A Theory of Text Organization. *TEXT* 8(3), 243-281.

Marcu, D. 1997. The rhetorical parsing of natural language texts. Proc. of the 35th Annual Conference of the ACL, 96-103.

Marcu, D.; Amorrortu, E. and Romera, M. 1999. Experiments in Constructing a Corpus of Discourse Trees. In: Proc. of ACL Workshop 'Towards Standards and Tools for Discourse Tagging', University of Maryland.

Miltsakaki, E.; Prasad, R.; Joshi, A. and Webber, B. 2004. Annotating Discourse Connectives and their Arguments. In: Proc. of the HLT/NAACL Workshop 'Frontiers in Corpus Annotation', Boston.

Pasch, R; Brausse, U.; Breindl, E. and Wassner, H. 2003. *Handbuch der deutschen Konnektoren.* Berlin: deGruyter.

Schilder, F. 2002. Robust Discourse Parsing via Discourse Markers, Topicality and Position. *Natural Language Engineering* 8 (2/3).

Stede, M. 2004. The Potsdam Commentary Corpus. In: Proc. of the ACL Workshop 'Discourse Annotation', Barcelona.

Sumita, K.; Ono, K.; Chino, T.; Ukita, T.; Amano, S. 1992. A discourse structure analyzer for Japanese text. Proc. of the International Conference on Fifth Generation Computer Systems, 1133-1140.

Webber, B.; Knott, A.; Stone, M. and Joshi, 2003. A. Anaphora and Discourse Structure. *Computational Linguistics* 29(4), 545-588.