

# Processing Text-Technological Resources in Discourse Parsing

Henning Lobin, Harald Lüngen, Mirco Hilbert, and Maja Bärenfänger

POSTPRINT

**Abstract.** Discourse parsing of complex text types such as scientific research articles requires the analysis of an input document on linguistic and structural levels that go beyond traditionally employed lexical discourse markers. This chapter describes a text-technological approach to discourse parsing. Discourse parsing with the aim of providing a discourse structure is seen as the addition of a new annotation layer for input documents marked up on several linguistic annotation levels. The discourse parser generates discourse structures according to the Rhetorical Structure Theory. An overview of the knowledge sources and components for parsing scientific journal articles is given. The parser's core consists of cascaded applications of the GAP, a *Generic Annotation Parser*. Details of the chart parsing algorithm are provided, as well as a short evaluation in terms of comparisons with reference annotations from our corpus and with recently developed systems with a similar task.

## 3.1 Introduction

Relational discourse theories like RST [*Rhetorical Structure Theory*, 29, 31], D-LTAG [*Lexicalized Tree-Adjoining Grammar for Discourse*, 43], ULDM [*Unified Linguistic Discourse Model*, 33, 34], or SDRT [*Segmented Discourse Representation Theory* 2, 3] provide text-type independent principles for analysing coherence relations between parts of a text of different sizes. For some of these theories, discourse parsers have been implemented, notably for RST. Two features of RST make it especially favourable for an automatisisation of discourse analyses: RST utilises trees (not graphs like SDRT) as a data structure for discourse representation.

---

Henning Lobin · Harald Lüngen · Mirco Hilbert · Maja Bärenfänger  
Applied and Computational Linguistics, Justus-Liebig-Universität Gießen,  
Otto-Behaghel-Straße 10D, D-35394 Gießen, Germany  
e-mail: {henning.lobin, harald.luengen, mirco.hilbert,  
maja.baerenfaenger}@germanistik.uni-giessen.de

(It is in fact controversial whether graph-based representations are actually necessary for the representation of discourse structures [cf. e.g. 12].) And while in the definition of rhetorical relations in the original theory, references to the beliefs and intentions of speakers and hearers abound [cf. 29]), in the different approaches to RST-based discourse parsing it has been shown that automatic discourse analysis can also be achieved by applying mainly surface-oriented discourse markers (*cues*) [cf. 31, 36, 21]. In the following, we give a brief overview of previous RST approaches to discourse parsing.

[30, 31] presented several alternative algorithms for the RST parsing of unrestricted texts. One prerequisite for rhetorical parsing formulated by Marcu is the *compositionality principle* for RST structures, which states that a rhetorical relation holding between two text constituents (*spans*) also exists between their respective most salient subconstituents. According to Marcu, another prerequisite for the parsing of unrestricted texts is a feature-based description of discourse markers based on an extensive corpus analysis. Discourse markers are utilized both for the segmentation and identification of related discourse units and for the assignment of a particular rhetorical relation.

Corston-Oliver's [8] automatic Rhetorical Structure Analyser RASTA bases its rhetorical analyses on fully-fledged syntactic analyses of the sentences of a text (in this case articles from the Microsoft Encarta Encyclopedia). Thus the cues that indicate rhetorical relations comprise discourse connectives as well as syntactic and morphological features. A further novelty introduced in this approach is the association of *cue:relation* pairs with weights that are based on linguistic intuition. They are used to build up more plausible discourse representations before less plausible ones.

An extension of Corston-Oliver's algorithm is the symbolic RST parser for English developed by Le Thanh [20, 21, 23, 22]. It performs an automatic discourse segmentation into elementary discourse units, sentence-level discourse parsing using syntactic information and cue phrases, and finally, text-level discourse parsing using a beam-search algorithm. To reduce the search space, heuristic scores are used as constraints on textual adjacency and textual organisation. The parser was evaluated on a test corpus from the *RST Discourse Treebank* [7].

As an alternative approach [35, 36] implemented discourse parsing according to RST as a quantitative approach as a series of text classification decisions. Classification instances from a training corpus are represented as feature vectors and associated with rhetorical relation schemata. The linguistic features used include the occurrence of discourse markers in certain segment positions, concepts introduced by definite noun phrases, punctuation, POS tagging and lexical similarity [35]. Support vector machines (SVM) are used as a classification algorithm. As a representation format for RST trees, the XML application URML [*Underspecified Rhetorical Markup Language*, cf. 37] is chosen. In an URML document, alternative tree structures can be presented as well. Besides, URML is used to represent partial results in the parsing process, similar to a chart in chart parsing.

URML is also used in the approach by [14], employing a feature-based RST grammar with a rule hierarchy. The grammar also includes robust rules that

**Table 3.1** Annotations in the SemDok corpus.

| XML annotation layer   | # annotated articles |
|--|----------------------|
| Logical document structure (DOC)   | 47+2                 |
| Morphological and syntactic structure (CNX)<br>(using the tagger <i>Machinese Syntax</i> from Connexor Oy) | 47 +2                |
| Discourse markers (DMS)  | 47                   |
| Rhetorical structure (RST-HP)  | 5+2                  |
| Discourse segments (SEG)   | 5+2                  |
| Anaphoric structure (CHS) (from Sekimo project)  | 3+2                  |
| Lexical chains (LC) (from HyTex project)   | 1+2                  |
| Genre-specific text type structure (TTS)   | 47                   |

combine subtrees when no discourse marker is found. Using this grammar, a standard chart parsing algorithm is applied for discourse parsing. The chart is extended to accommodate parse forests, and URML is used for their representation.

In several of the above described projects, collections of newspaper articles were used as test corpora [36, 38, 19]. The goal of the SemDok project was to design and implement a new RST parser for the text type<sup>1</sup> of (German) *scientific journal articles* as a text-technological application.<sup>2</sup> Scientific articles form a more complex text type than newspaper articles – primarily due to their deeply nested logical document structure. A discourse parser therefore has to resolve a higher number of potential relational combinations of text segments. Besides the traditional discourse markers such as lexical cues, grammatical features and punctuation, features derived from analyses of text and document structures need to be included as well. An overview of the requirements for such an approach in terms of linguistic foundations, resources, and an application scenario is given in [27]. The linguistic resources are made available for the SemDok parser using text-technological (XML-based) standards, formalisms, methods and tools and are described in the following sections.

## 3.2 Corpus

A corpus of German linguistic journal articles, which was created between 2002 and 2008, served as a development corpus. It provides XML annotations on various linguistic and text-structural analysis layers. The corpus contains 47 German articles of the online journal *Linguistik Online* ([www.linguistik-online.de](http://www.linguistik-online.de)) from between 2000 and 2003 (comprising approx. 360,000 word forms). One newspaper article (from the German weekly *Die Zeit*) and one web-published article on

<sup>1</sup> The term *text type* is used as an equivalent for *genre*.

<sup>2</sup> SemDok was a project within the DFG research group 437 *Text-technological modelling of information*. The discourse parser was developed in the project's second funding phase (2005-2008) called *Generic document structures in linearly organised texts*.

hypertext were added (from the corpora compiled in the projects *Sekimo*<sup>3</sup> and *HyTex*<sup>4</sup>, cf. “+2” in Table 3.1).

An overview of the SemDok corpus and its different annotation layers is given in Table 3.1. The different annotation layers were added according to the framework of *XML-based multi-layer annotation* [44]: Each annotation layer is stored in a separate XML document, i.e. the primary (text) data are copied several times. Since in all annotation layers, the primary data are absolutely identical, relations holding between elements on different annotation layers can be analysed using the so-called *Sekimo tools*<sup>5</sup> [45].

The corpus was semi-automatically annotated according to a modified Doc-Book format (“DOC”, [cf. 41, 24]). For the annotation of morphology and syntax (“CNX”) we employed the commercial software *Machineese Syntax* from Connexor Oy. Machineese Syntax yields dependency trees according to the *Functional Dependency Grammar* [FDG, 39] as an XML-like annotation. For an annotation of lexical discourse markers (“DMS”), a tagger was developed which basically performs lexical insertions according to the SemDok discourse marker lexicon in combination with some context checking [cf. 27]. The initial discourse segmentation (“SEG”) was achieved by a segmentation program also developed in SemDok, which segments a document according to criteria on punctuation, grammar, and logical document structure [cf. 25]. Anaphoric structure (or referential structure) represented by the annotation layer “CHS”, was added to the SemDok documents corpus in the

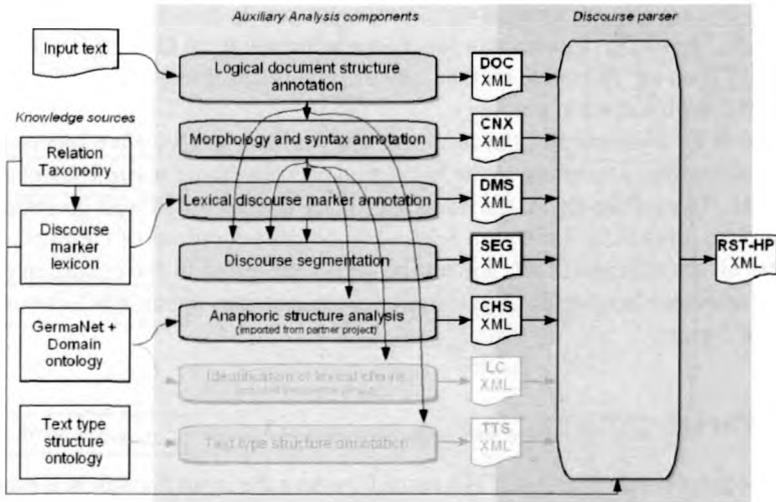
```
<para xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="hypo-para.xsd" relname="Contrast" id="i12">
  <n id="i13">
    <hypo id="i14" relname="Elaboration-example">
      <n id="i1">
        <t id="ti1">In der Schrift hat die Sprachpflege einen etwas besseren
          Erfolg als im Gespräch gehabt.</t>
        </n>
        <s id="i2">
          <t id="ti2">In öffentlichen Dokumenten ist man z.B. darauf bedacht,
            dass die Termini dem Gebrauch in Schweden entsprechen.</t>
          </s>
        </hypo>
      </n>
      <n id="i4">
        <t id="ti4">Trotzdem enthalten sowohl Sachtexte als auch die
          Belletristik
          sprachliche Züge, die den Schweden fremd vorkommen.</t>
        </n>
      </para>
```

**Listing 3.1** RST analysis in RST-HP.

<sup>3</sup> <http://www.text-technology.de/Sekimo/>

<sup>4</sup> <http://www.hytext.info>

<sup>5</sup> The *Sekimo tools* operate on a stand-off Prolog fact base format and include tools for merging different annotation layers, for transforming between XML and the Prolog format and for checking relations holding between elements of single annotations layers.



**Fig. 3.1** Discourse Parsing Architecture.

partner project Sekimo [cf. 10]. The SemDok parser is also projected to handle two further annotation layers: an annotation of lexical chains (“LC”), as provided by the partner projects HyTex [cf. 9] and IndoGram [cf. 42], and an annotation of the genre-specific text type structure (“TTS”), available from the first funding period of SemDok [cf. 4].

Apart from the above annotation layers, which serve as auxiliary annotations in the discourse parsing process, several corpus articles were also provided with reference annotations according to the Rhetorical Structure Theory in RST-HP. RST-HP is an XML application developed in the SemDok project to represent RST trees in XML [27, 17]. “HP” stands for `<hypo>` and `<para>`, which are two element types to label RST subtrees as either hypotactic or paratactic (relations). An example of an RST analysis in RST-HP is given in Listing 3.1. RST-HP is also the target format of the SemDok discourse parser.

### 3.3 Architecture

For the technical realisation of our discourse parsing approach we chose a pipeline architecture, in which linguistic analyses of one text document at different linguistic levels are provided by auxiliary analysis components (preprocessors) implemented as part of the SemDok project, or by project-external software. These components were also used in creating the SemDok corpus (Section 3.2).

The pipeline architecture is shown in Figure 3.1. The auxiliary analysis components are shown in the middle part, on the left-hand side there are four knowledge

sources that are used by some of these preprocessors. In this chapter, we do not discuss these knowledge sources, but see [6] for information on the *Relation Taxonomy RRSet*, [27] on the discourse marker lexicon, and [26] for techniques of combining GermaNet with a domain ontology.

As well as in the corpus, the results of the auxiliary analyses are represented as XML annotations according to the principles of *XML-based multi-layered annotation* [44]. To evaluate them, the SemDok parser was provided with an interface to the Sekimo tools [45]. Using the Sekimo tools, configurations of elements and attributes on the different XML annotation layers are tested in the condition parts of reduce rules (see Section 3.4.3.2). Finally, the parser puts out an XML document in RST-HP format.

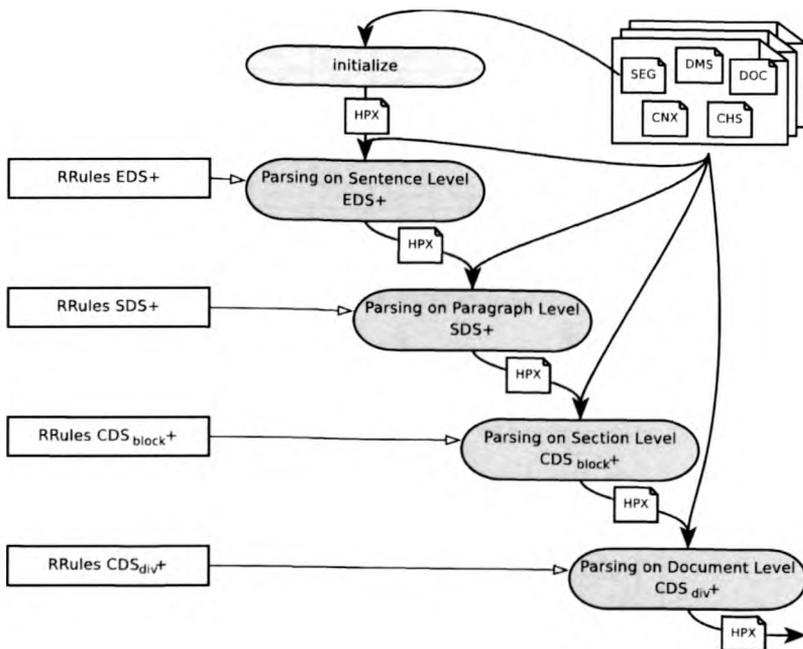
### 3.4 Parser

The discourse parser is realised as a cascade, where the input document is processed in bottom-up fashion along its document levels (see Figure 3.2). The information about document levels is provided by the annotation layer SEG, which contains the initial segmentation into elementary and complex discourse segments [cf. 25]. SEG is the only obligatory annotation layer, i.e. all other layers are optional in terms of input requirements.

In each cascade step, a parsing iteration is activated for the associated document level. At the first level, called “EDS+”, for each containing element of type “SDS” (sentential discourse segment), “EDSes”, (elementary discourse segments – mostly clauses) are recursively combined to form structures on SDS level. At the second level, called “SDS+”, SDSes are combined to form complex discourse segments on block level (mostly paragraphs, “CDS<sub>block</sub>”), on the third level, the block-level segments are combined to form CDSes of type “division” (sections and the like, “CDS<sub>div</sub>”), and finally, the CDS<sub>div</sub> are combined up to the level of the complete document (i.e. one CDS<sub>doc</sub>). In each cascade step, the respective higher level elements are called the *containing elements*, they thus act as top-down constraints in the otherwise bottom-up parsing process.

#### 3.4.1 Initialisation

The chart parser’s internal representation “HPX” [cf. 17] is an extended version of the target format RST-HP that can accommodate underspecified information, cf. Section 3.4.3. During the parsing process, the HPX chart is gradually augmented by new information in the form of new chart edges. Hence, in an initialisation phase, the elementary discourse segmentation SEG is converted to HPX by changing each EDS into a terminal edge element <t> and assigning it an underspecified nuclearity in the form of an <undefined> element [cf. 31, 152ff]. That way, a sequence of adjacent <undefined> elements forms the basis of the parsing process.



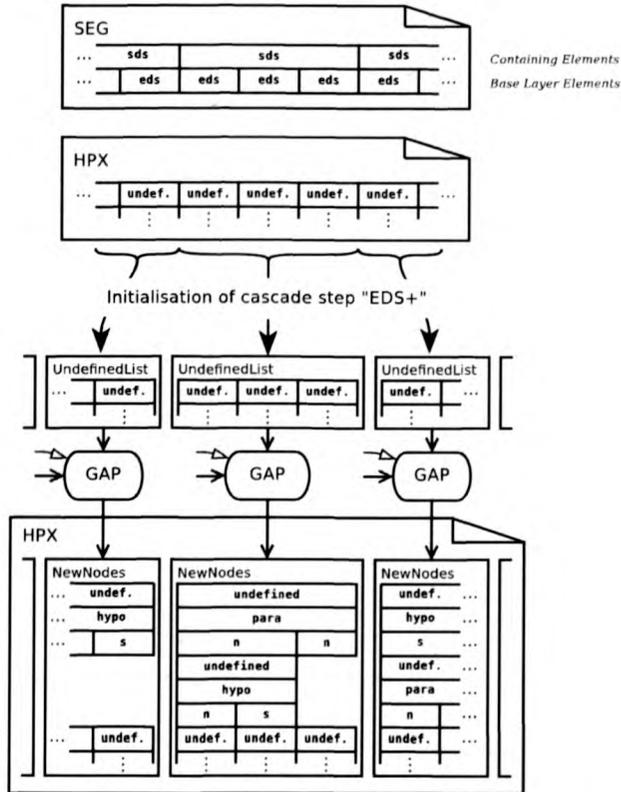
**Fig. 3.2** Parsing Cascade.

The actual parsing is achieved by consecutive calls of the central parsing component *GAP* (*Generalized Annotation Parser*) for each containing element on the SEG layer. The GAP is described in Section 3.4.3.

### 3.4.2 Cascade Step

An example of the operation of the algorithm in one cascade step is illustrated in Figure 3.3. The invocation of a cascade step depends on several parameters.

- The *Base Layer Type* and the *Containing Layer Type* specify the document level to be parsed. On the level of “EDS+”, for example, RST structures are built whose leaf nodes correspond to segments of the base layer type “EDS” and whose root node completely covers the associated segment of the containing layer type “SDS”.
- The *Reduce Rule Set* is the third mandatory parameter. It contains the rules stating how a set of adjacent segments may be linked by a rhetorical relation and combined to form a bigger segment.



**Fig. 3.3** Initialisation and execution of one cascade step by the example of the document level "EDS+" (*Sentence Level*).

Two further parameters modify and extend the effect of the reduce rules: the *Rhetorical Relation Set* and the *Default Relation*.

- The SemDok *Rhetorical Relation Set* (RRSet) forms a taxonomy of rhetorical relations suitable for our application purpose [cf. 6]. A full set consisting of 44 relations and a reduced set consisting of 30 relations with less specific relations at the leaves of the taxonomy are available. Depending on the RRSet variant selected for a cascade step, the reduce rules are customised to label a combined segment with a more specific (e.g. ELABORATION-CONTINUATION-OTHER) or a more general relation (e.g. ELABORATION).
- In the *Default Relation* parameter, a relation is specified that combines two adjacent segments when no regular reduce rule matches the configuration and its features. As values, the three relations most frequently found in the SemDok corpus can be specified: the general multi-nuclear coordination relation LIST-COORDINATION, and two multi-nuclear relations indicating thematic progression, ELABORATION-DRIFT and ELABORATION-CONTINUATION-OTHER.

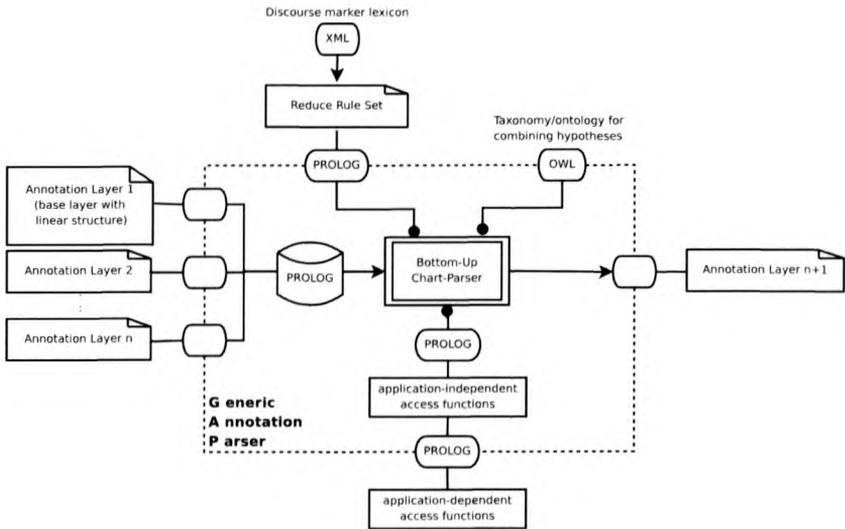


Fig. 3.4 GAP.

Within a cascade step, a list of `<undefined>` elements is generated first for each segment of the *Containing Layer Type* (*Containing Element*). Each `<undefined>` element corresponds to an element of the *Base Layer Type* (*Base Element*) (for an example cf. the initialisation of cascade step “EDS+” in Figure 3.3).

Iterating over all *Containing Elements*, the GAP is then invoked for each list with multiple `<undefined>` elements.<sup>6</sup>

The GAP attempts to build a partial HPX chart which spans the current *Containing Element*, applying the reduce rules, which make reference to information available on the auxiliary annotation layers. If the HPX chart built fails to completely span the current *Containing Element*, an artificial “RST collection” edge is inserted, which spans the current *Containing Element* by connecting a sequence of longest adjacent `<undefined>` elements in the `<undefined>` list. This ensures an output of connected RST trees and robustness of discourse parsing.

### 3.4.3 Generic Annotation Parser (GAP)

The Generic Annotation Parser (GAP), as displayed in Figure 3.4, has been conceived as an abstract parsing system augmenting  $n$  input annotation layers with identical primary text data by one new,  $n + 1$ st output annotation layer representing a constituency structure. The building of the output annotations is based on the reduce rule set that contains reduce rules to add new annotations starting from the

<sup>6</sup> The GAP is not invoked for lists with only one `<undefined>` element, as that already covers the respective *Containing Element*.

base annotation layer. In the reduce rules, reference can also be made to information on further input annotation layers.

The base annotation layer must consist of a sequence of adjacent elements that covers the whole text and provides the basis for building the new, output structure. In the case of discourse parsing in SemDok, these are the `<undefined>` elements on the annotation layer HPX, which can be changed into `<n>` or `<s>` elements as exemplified in Figure 3.3.

The rule set contains declarative reduce rules which are selected based on the sequences (mostly pairs) of matching elements on the base layer and on additional conditions referring to other annotation layers.

Since element configurations may be ambiguous i.e. match with multiple rules and associated relations, the quality of each possible output structure is evaluated by means of a scoring function. Scoring is discussed more fully in Section 3.4.3.6, further details of the reduce rules are given in Section 3.4.3.2.

In the following section, we describe the bottom-up passive chart parser that constitutes the core of the GAP.

### 3.4.3.1 Chart Parser

A discourse parser needs a strategy to handle ambiguities, arising e.g. due to discourse markers which are ambiguous according to the discourse relation they indicate, or their scope. Chart parsing is designed to efficiently store and retrieve information about partial structures that have already been parsed [cf. 16].

The chart of the SemDok Parser is defined by minimally extending the Sekimo fact base format [45] for RST-HP-compliant XML documents such that SemDok chart edges are always potential Prolog node/5 facts representing the XML elements of a RST-HP result document. The chart format is therefore called HPX, for *HP extended*. An HPX chart edge has the following components:

1. Layer: name of the base Layer; in the discourse parsing application: HPX
2. Start: PCDATA offset of the begin of the text range spanned by the chart edge (XML element)
3. End: PCDATA offset of the end of the text range spanned by the chart edge (XML element)
4. NodeID: edge ID of the chart edge (node ID of the XML Element)
5. LoLoI: *List of Lists of child node IDs*, pointers to the child edges
6. Score: score according to probabilistic parsing, cf. Section 3.4.3.6
7. DM-ID-List: list of those discourse markers that have already been used in the parsing history, cf. Section 3.4.3.7
8. Element: XML element name; in the application of discourse parsing, one of `n|s|para|hypo|embed|t|undefined|rstCollection`

The algorithm we employ for discourse parsing is based on *bottom-up passive chart parsing* [cf. 32, 1]. In our variant, no edge sequence is analysed more than

once, because in the loop over the current set of edges, each edge is checked only against those right-adjacent edges that have been inserted in the previous loop.

Node sequences of  $n$  nodes (as opposed to node pairs of two nodes) are currently analysed in two reduce rules for recognising embedded discourse segments (<embed> constructions cf. [27]).

### 3.4.3.2 Rule Components

#### Reduce Rules

For each cascade step (EDS+, SDS+, CDS<sub>block</sub>+, and CDS<sub>div</sub>+), an individual set of reduce rules is loaded. The reduce rules presently used in the SemDok parser have been acquired in two ways:

1. rules generated from the SemDok Discourse Marker Lexicon [27] using an XSLT style sheet
2. rules manually encoded in Prolog

Naturally, the rules which were generated from the Discourse Marker Lexicon mainly refer to the annotation layer DMS (cf. Section 3.2). The manually encoded rules typically refer to other layers and are based on findings from qualitative and quantitative corpus analyses, e.g. rules for assigning the ELABORATION relation and several of its subtypes which refer to the CHS layer as described in [5].

Listing 3.2 shows the entry for *dagegen* (“in contrast”) in the SemDok Discourse Marker Lexicon. Listing 3.3 shows the Prolog reduce rule generated from it.

The “corpus score” specified in the attribute @corpusScore is derived from an XML database consisting of 564 RST tree instances from an annotated subcorpus, in which the discourse markers that indicate the relation are annotated as well. The corpus score in Listing 3.2 specifies that 56% of all occurrences of *dagegen* in the subcorpus were marked to indicate the relation CONTRAST-MULTI. The simple score in the attribute @score, on the other hand, represents the a priori probability

```

<dm id="c333" typ="lexical">
  <cue>
    <text>dagegen</text>
    <lemma pos="ADV">dagegen</lemma>
    <position>
      <vorfeld>+</vorfeld>
      <mittelfeld>+</mittelfeld>
    </position>
  </cue>
  <rels default="Contrast-multi">
    <relation corpusScore="0.55556" score="1" relname="Contrast-multi"
      skopus="sds+" typ="n" beds-richtung="1"/>
  </rels>
</dm>

```

**Listing 3.2** Entry in the discourse marker lexicon.

```

-----
% Case sentence adverb/coordinating conjunction "dagegen"
% Type N-N with simple lexical DM in N2

reduce_rule(dm1ist1, [N2, N1|[]], [Np|[]], L_new_undefined) :-

    gap_baselayer(BaseLayer),

    node(BaseLayer, _Start1, _End1, N1, _, _Score1, _DML1, element('
        undefined')),
    node(BaseLayer, Start2, End2, N2, _, _Score2, DML2, element('undefined')
    ),

    % Constraint:
    one_relation(inclusion_B_in_A, 'undefined', BaseLayer, N2, dm, 'DMS',
        N_d, Start2, End2, Start_d, End_d),

    % Constraint:
    attr('DMS', Start_d, End_d, N_d, 'lemma', 'dagegen'),
    attr('DMS', Start_d, End_d, N_d, 'pos', 'ADV'),

    % Get DMID and check DMID in DML2:
    attr('DMS', Start_d, End_d, N_d, 'id', ID),
    nonvar(ID), nonvar(DML2),
    not(member(('DMS', ID), DML2)),

    reduce_to_N_N(N1, N2, 'Contrast-multi', Np, L_new_undefined, 0.00829, [(
        'DMS', ID)]).

```

**Listing 3.3** Reduce in Prolog, generated from the discourse marker lexicon.

of the discourse marker derived from its ambiguity in the Discourse Marker Lexicon. In the entry in Listing 3.2, the simple score is 1, because in the lexicon the discourse marker is specified to indicate only one relation.

A reduce rule as in Listing 3.3 is processed in the following way: First, all components of the current node sequence are retrieved from the chart via their IDs. (In the example in Listing 3.3, the variables *N1* and *N2* represent the IDs of the current node sequence.) Then it is checked whether an occurrence of `<dm>dagegen</dm>` on the annotation layer 'DMS' is included in the text span of node *N2* on the base layer. Finally, it is checked whether the discourse marker has not already been used in the parsing history of the segment represented by *N2* (using the DM-ID (discourse marker identifier) list *DML2*). When all constraints are satisfied, the required components will be passed to the reduce schema `reduce_to_N_N` which is invoked to insert the edges representing a new multi-nuclear relation into the chart.

The generated score of the rule (0.00829) is not identical with the conditional probability in the `@corpusScore` attribute in the discourse marker lexicon entry in Listing 3.2, since it has been combined with the a priori probability of the relation `CONTRAST-MULTI` (also acquired from the corpus) when generating the Prolog rule.

## Reduce Schemas

When a reduce rule and its constraints match XML annotations of the current node sequence, the edges forming a new RST subtree are inserted into the chart. For this purpose, five reduce rule application schemata are available (similar to the RST application schemata in [29]). Depending on the schema, either a `<hypo>`, a `<para>` or an `<embed>` edge is generated, as well as the corresponding `<n>` and `<s>` edges, and one new `<undefined>` edge, which will be available in the subsequent parsing process. The following schemas are available.

1. `reduce_to_N_S`: mono-nuclear schema
2. `reduce_to_S_N`: mono-nuclear schema
3. `reduce_to_N_N`: bi-nuclear schema
4. `reduce_to_N_N_List_Add`: schema for a “tree-adjoining” construction of multi-nuclear structures. Proper multi-nuclear structures (with more than two nuclei, such as potentially occurring with the relations `LIST`, `SEQUENCE`, and their subtypes) will first be initialised by an application of Schema 3 to the first two nuclei. The remaining nuclei will then be added by iteratively applying Schema 4 in the parsing loops to follow (when the constraints of the rule match). This procedure represents one way to derive multinuclear structures using only binary rules. Within Schema 4, incomplete intermediate multinuclear structures are removed from the chart; this is the only possible destructive action during chart building. Nevertheless, some incomplete (from the viewpoint of a correct reference annotation) multi-nuclear structures may still be kept in the chart, since Schema 3 may be applied to non-initial elements of a multi-nuclear structure as well.
5. `reduce_to_embed`: schema for embedded satellite constructions. This schema has actually two components, one for two embedded satellites, and one for three embedded satellites within a nucleus. As `<embed>` constructions can only be built over EDSes, they could alternatively be parsed in a separate function to be invoked before the first call of the GAP. The chart parsing algorithm would then operate on node pairs instead of node sequences, because all the remaining reduce rules in the SemDok parser are binary.

### 3.4.3.3 Ranking of Reduce Rules

In the SemDok parser, each rule is ranked so that rules are grouped into specific ones, less specific ones, and default rules. The parser tests a node sequence against rules in a group with a more general rank only when no rules in the more specific groups matched. Rule ranking allows for a prioritisation of rules. Another method of prioritising rules employed in the SemDok parser is scoring (cf. Section 3.4.3.6). The two methods supplement one another as ranking represents a discrete gradation of rules that may lead to an absolute exclusion of certain rules, as opposed to scoring which represents a continuous measurement of the quality of parse trees. In the

current version, three named ranks are defined, representing the following groups of rules.<sup>7</sup>

1. ‘dmlist1’: Rules based on lexical discourse markers. If one of these matches, do not continue with the following rule groups.
2. ‘elab’: Rules based on the annotation of anaphora, mostly indicating ELABORATION. If one of these matches, do not continue with the following rule group.<sup>8</sup>
3. ‘list’: A group containing one default rule.

#### 3.4.3.4 Supplementary Functions

To check the constraints formulated in the reduce rules a set of *application-independent functions* (i. e. Prolog predicates) is available. It contains predicates for querying information about the general configuration of XML elements on the multiple annotation layers as stored in the Prolog fact base, such as “Are the nodes in  $L$  children of node  $N$ ?”, or “Are the nodes in  $L$  all the children of node  $N$ ?”

Furthermore, a set of *application-dependent functions* (i. e. discourse parsing-specific predicates) is available. For the most part, it contains query predicates referring to the grammatical annotation, such as: “Does segment  $N1$  correspond to a complete sentence?”, “Is discourse marker  $D1$  contained in the first sentence of  $N2$ ?” or “Is the anaphoric expression  $A1$  the subject of the first sentence in  $N2$ ?”

The purpose of separating application-independent and application-dependent functions is to be able to simply exchange the module containing the application-dependent functions when the GAP is employed for a different application such as sentence parsing.

#### 3.4.3.5 Node Packing

A situation where the parser finds two or more analyses for the same local text range is called *local ambiguity*. Local ambiguity occurs when more than one reduce rule is applicable to an edge sequence, or because a discourse marker is ambiguous, or because the current segments contain several discourse markers indicating different rhetorical relations for the combination of the segments, or because  $n$  different chart edge sequences were combined to form  $n$  new chart edges with identical range. Local ambiguity has to be distinguished from cases where the current segments contain several discourse markers indicating *the same* rhetorical relation for the combination of the segments, these are treated by an adjustment of the score, cf. Section 3.4.3.6.

In the original chart parsing algorithm, when  $n$  analyses are found over the same text span,  $n$  `<undefined>` edges will be inserted in the chart, and the overall

<sup>7</sup> Not all ranks are used for each document level/in each cascade step.

<sup>8</sup> Rules in which anaphora indicate (a type of) ELABORATION have a default character with respect to rules based on lexical discourse markers, cf. [6].

ambiguity grows exponentially. To eliminate this kind of ambiguity, we use *packed representations* in the line of [40]:<sup>9</sup>

For a set of edges with an identical range (start and end offsets) but different analyses (their relation labels or sets of child edges), only one new `<undefined>` edge will be inserted into the chart. Note that in discourse parsing, the type of an edge (the RST tree type plus its relation label) is irrelevant for its combination with another segment in a rule application, and consequently edges can be packed regardless of their type. In syntax parsing, in contrast, no edges with identical range but different types (e.g. AP, NP, VP) may be packed.

#### 3.4.3.6 Scoring

The sixth argument of a chart edge is its *score*, which assigns a rating to the RST tree it represents. Its purpose is to make competing hypotheses of rhetorical relations comparable. The score of an edge depends on the context in which it is inserted into the chart. In principle, we distinguish two cases in the calculation of a new score. When two or more adjacent discourse segments are combined to form a larger segment (case “children2parent”), a new score is computed for the edge representing the larger segment [cf. 28, 23]. Similarly, when several alternative analyses are combined in a packed edge (case “alternative”, cf. Section 3.4.3.5), a new, averaged score is computed for the packed edge. For both cases of score combinations, a number of different mean calculations have been implemented: product, geometric mean, arithmetic mean, quadratic mean, and maximum of the involved scores.

Calculating the product is the common method to combine two independent probability values. [28] alternatively suggested the geometric mean in order to reduce the influence of very low partial scores on the total score, e.g. on account of very few occurrences of a discourse marker in a corpus. The arithmetic mean is the classic average calculation, treating each partial score equally. Using the square mean, good scores have a higher influence on the resulting score than bad ones in comparison with the arithmetic mean. The maximum of the underlying scores is another possible heuristic for the combination of alternative analyses, meaning that a packed edge will get the score of the best-scored edge among the alternatives it represents.

A score newly computed using one of these methods is set off against the score of the rule that has been applied for inserting the edge. The rule score is the a priori probability of the rhetorical relation propagated by the rule combined with the conditional probability of the relation given the discourse marker that was tested in the rule. The probabilities used have been estimated by calculating the percentages of relation occurrences and discourse marker occurrences in the SemDok corpus (cf. Section 3.4.3.2).

Which one of the mean calculations is to be chosen in the two cases can be parametrised in the main call of the SemDok parser, so that the best settings for the parameters can be determined in test runs.

<sup>9</sup> Packed representations were originally introduced as *shared forests of parse trees*, as [40] is not a chart parsing approach.

A special case occurs when several reduce rules indicate one and the same rhetorical relation for the same segment combination, which means that a matching rule would lead to a set of new edges that are already in the chart. In that situation, no new RST tree is inserted in the chart but the score of the existing chart edges is updated by increasing it by the rule score of the newly matched rule. That way, the score for a relation will be rated higher, the more cues for the relation are found.

### 3.4.3.7 DM-ID Lists

The seventh component of a chart edge is a list of identifiers (i.e. values of XML ID attributes of XML elements representing the discourse markers in the linguistic annotations of the input document) of those discourse markers that were used in the parsing history of the RST tree represented by the edge, cf. Section 3.4.3.1. In the derivation of *one* RST tree, *one* discourse marker must have induced exactly *one* relation. Hence, one has to keep account of those discourse markers that have already led to rule applications in the bottom-up parsing process. This purpose is served by the DM-ID list. The set of already consumed discourse markers is indicated on the chart edges of the types *para*, *hypo*, or *embed*. When a new edge is inserted in the chart as the result of a successful rule application, the DM-ID list of the edge for the new RST tree edge consists of the union of the DM-ID lists of its child edges plus the DM-ID(s) of the discourse marker(s) that matched in the current rule application.

A conflict occurs when *packed edges* (cf. Section 3.4.3.5) are inserted in the chart. Like all chart edges, packed edges are specified for exactly one DM-ID list. However, they do not represent one unique, but several tree derivations. Thus, if *A* and *B* are two derivations packed in one packed edge, in derivation *A*, a discourse marker may have been applied that has not been applied in derivation *B*. To capture all the possible cases, one would have to employ disjunctions of DM-ID lists, which would of course counteract the purpose of packed representations. Hence, in the SemDok parser, the following three heuristics are implemented for combining two DM-ID lists: intersection, union, and the so-called majority union.

When intersection is chosen as the combination method, some DM-IDs may get lost so that the associated discourse markers might (falsely) be applied once again in the subsequent parsing process. When the DM-ID lists of the edges  $K_1, K_2 \dots K_n$  to be packed are combined by the union operation, the packed edge does represent the derivation associated with the edge  $K_i$ , but its DM-ID list possibly also contains discourse markers of the derivation associated only with the edge  $K_j$ . In the subsequent parsing process, these will (falsely) not be available any more for the continued derivation associated with  $K_i$ . The “majority union” of  $K_1, K_2 \dots K_n$  is defined such that the DM-ID list of the packed edge will contain only DM-IDs contained in least 50% of the DM-ID lists associated with  $K_1, K_2 \dots K_n$ . It is then possible that some discourse markers may be falsely applied a second time in the subsequent parsing process, however in less cases than with regular list intersection. It is also possible that some discourse markers are falsely not available anymore, however in less cases than with regular list union.

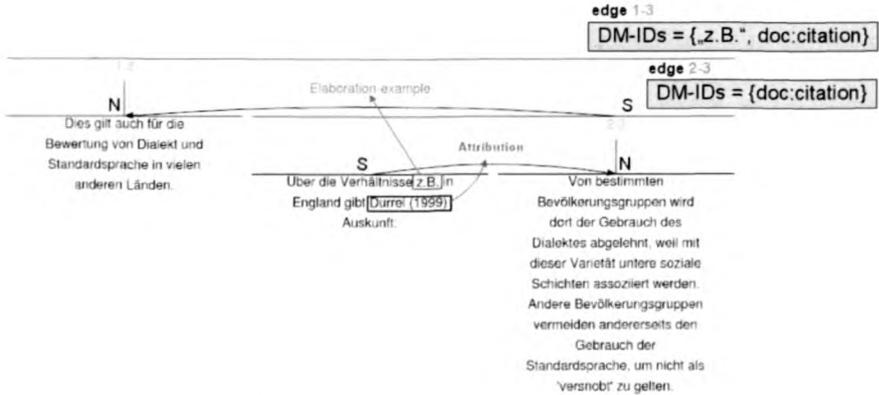


Fig. 3.5 Consumed discourse markers in the DM-ID list.

When calling the SemDok parser, one of the three methods to combine DM-ID lists for packed chart edges needs to be specified. Since node packing itself can also be selected or deselected on the parser's top-level call [cf. 18], it can be determined in evaluation suites which of the three methods yields the best results and how high the error rate is in comparison with parsing without node packing, i.e. with unmanipulated DM-ID lists.

### 3.4.3.8 Relaxation of the Compositionality Criterion

In [31], a *compositionality criterion* for rhetorical structures is formulated stating that a relation that holds between two text segments also holds between at least two of the nuclei among their embedded segments. Consequently, according to Marcu, in discourse parsing it is sufficient to consider discourse markers only in the top nuclei of the segments concerned. However, numerous counterexamples to this claim can be found in the SemDok corpus, one of which is illustrated in Figure 3.5. Between segment 2 (the satellite) and segment 3 (the nucleus) in the lower RST subtree, an ATTRIBUTION relation holds. This relation is indicated by a citation marker (XML element `<doc:citation>` on the annotation layer DOC) in segment 2, strictly speaking in combination with the colon and the predicate "gibt Auskunft" ("provides information"). In the top-level RST tree (segments 1-3) an ELABORATION-EXAMPLE relation holds, indicated by the discourse marker "z. B." ("e.g.") in segment 2, a lower-embedded satellite from the perspective of the top-level tree. Thus, in case of the ELABORATION-EXAMPLE relation in Figure 3.5, a consideration of the embedded nuclei is not sufficient because the relevant discourse marker occurs in the subordinated satellite. The observation that the compositionality criterion is not sufficient was made previously by [20], and many more examples

of this kind can be found in the SemDok corpus. Consequently, discourse markers occurring in *any* subsegment of the candidate segments are considered in the reduce rules for cascade step EDS+. For the higher segment levels SDS+, CDS<sub>block</sub>+, and CDS<sub>div</sub>+, however, lexical discourse markers are only considered in the first sentence (SDS) of the second segment, as, with the exception of list contexts, no instance of a lexical discourse marker indicating a relation and occurring in a non-first sentence of its second segment was found in the SemDok corpus.

### 3.4.4 Traversing the Chart

Our result chart is equivalent to a parse forest [cf. 40, 14], i.e. the chart edges represent the nodes of subtrees connected by the ID pointers in the edges' LoLoIs, representing sets of alternative child node lists (see Section 3.4.3.1). Starting from the "root edge" (the one that spans the complete document and becomes the root node of any result tree), the chart can be traversed along the LoLoIs to generate RST result trees. In the case of a packed edge, the scores of its sub-edges are used to select only the best-rated alternative(s).

Our chart traversal algorithm produces a set of  $x$  best-rated parse trees and stores each in a separate Prolog fact base. They can subsequently be exported into RST-HP XML documents using the Sekimo tool `prolog2xml`<sup>10</sup>.

The exact number of the best-rated parse trees cannot be determined beforehand but results from the number of alternative branches traversed on account of the scores found. To better be able to manipulate the number of result trees, it is intended to implement the possibility of specifying a relative threshold  $\vartheta$ , according to which alternatives can be selected by comparing it with the edge scores instead of automatically selecting all of the  $x$  best alternative edges.

For the visualisation and exploration of result trees, a graphical web interface was developed which displays an RST tree structure as well as the related document structure and the text, and also provides the possibility to navigate both structures and between both structures [cf. 18].

## 3.5 Evaluation

In a parsing experiment, six documents from our development corpus were parsed on the sentence and block level (EDS+ and SDS+). Two of them were not scientific articles, but one web-published article on hypertext and one newspaper article from our partner projects HyTex and Sekimo, cf. Section 3.2.

---

<sup>10</sup> <http://coli.lili.uni-bielefeld.de/Texttechnologie/Forschergruppe/sekimo/python/>

**Table 3.2** Comparative evaluation.

|                          | #relation set | precision | recall |
|--------------------------|---------------|-----------|--------|
| SemDok Exp 1 block level | 44 relations  | 11.53     | 32.71  |
| SemDok Exp 2 block level | 30 relations  | 12.22     | 36.64  |
| LeThanh 1 text level     | 22 relations  | 38.5      | 39.6   |
| LeThanh 2 text level     | 14 relations  | 39.3      | 40.5   |

The six articles contained between 1235 and 9988 wordforms, altogether 26325 wordforms. Their manually built reference annotations of their RST structure contained 2219 elementary discourse segments. The automatically generated initial segmentation was post-edited with respect to faulty segmentations that were introduced on account of errors in the morphology/syntax analysis but otherwise not adapted to the reference annotation.

In the evaluation, the parser’s output RST analyses were compared with the reference annotation. Only completely agreeing RST subtrees counted as a match, i.e. RST subtrees had to agree with respect to their relation label and the text range of the combined segment, the text ranges of the constituent segments and the nuclearity labels of the constituent segments.<sup>11</sup> We evaluated two parser runs on the six documents, one in which the full RRSet was used [cf. 6], and one in which a reduced RRSet with 30 (partly underspecified) relation labels was used. The results of the two runs are shown in the first two lines of Table 3.2.

Many relations present in the reference annotations are still not indicated by surface-related discourse markers as are currently analysed in the SemDok approach. Their absence is the main reason for the recall values of 32.71% and 36.64%. Note also that discourse analysis in terms of RST is not an unambiguously feasible task for human annotators, either. While producing the reference annotation of the corpus articles, our human annotators achieved agreements between  $\kappa = 0.47$  and  $\kappa = 0.81$  for RST analyses of the sentence and block level.<sup>12</sup>

The low precision values of 11.53% and 12.22% arise from the remaining ambiguities of many discourse markers with respect to the relation they indicate and to the scope of a relation, and also by a suboptimal performance of the export of the  $n$ -best subtrees from the chart using the scores in the traversal algorithm.

A recent symbolic rhetorical parser is the one by Le Thanh [20, 21, 23, 22, 19] for English, which represents an extension of Corston-Oliver’s [8] system. It was evaluated on a test corpus of 20 documents from the *RST Discourse Treebank* [7]

<sup>11</sup> We would like to thank Daniela Goecke of Bielefeld University for implementing the first version of the evaluation program.

<sup>12</sup> Three annotators annotated the same three corpus articles using the full RRSet consisting of 44 relation categories. This setting resulted in  $3 * 3 = 9$  agreement ratings. The number of RST subtrees of that annotator which had identified the most RST subtrees was taken as  $N$  in the  $\kappa$  formula. A match of RST subtrees (an agreement) was defined as explained for precision and recall above.

that contained between 30 and 1284 word forms [23]. This parser also performs an automatic segmentation of the input text into elementary discourse segments.

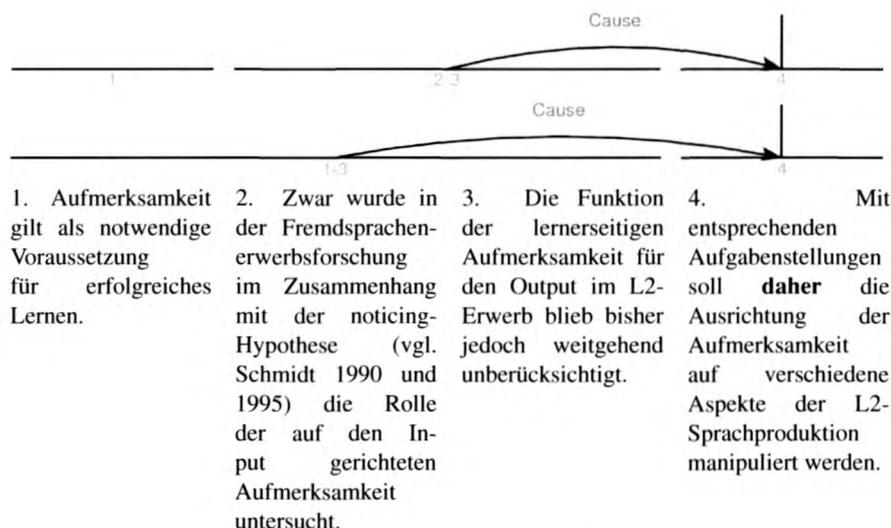
A comparison with this parser also suggests that recall values above 40% are generally hard to achieve on discourse segments that are bigger than sentences. The third and fourth line in Table 3.2 shows Le Thanh's [23] results for discourse parsing on the level of the entire text, which like the  $CDS_{block}$  level in the SemDok parser uses sentences as base segments. The results of her parser are better than those of the SemDok parser, but note that her texts are substantially shorter than the scientific articles of the SemDok corpus and that both versions of her relation set are substantially smaller.

### 3.6 Conclusion

In syntax parsing, the Earley algorithm [11], which combines the basic bottom-up approach with top-down predictions is usually applied to avoid a combinatorial explosion of the number of edges to be inserted in the chart. In discourse parsing, however, categorial top-down constraints corresponding to the phrase labels NP, AP, VP etc. are not available. Hence, in the SemDok approach, elements of the logical document structure of a text are used as top-down constraints by applying the central parsing component multiple times in a cascade for each containing element identified on the logical document structure annotation level of the input text. Further techniques implemented in the SemDok parser to reduce the hypothesis space are the representations of parse forest, node packing and a ranking of reduce rules. Moreover, rule scoring is applied during parsing and evaluated in the chart traversal to reduce the search space. Still the number of possible parses of the scientific articles of our corpus is quite high, as the precision figures of our evaluation indicate. Additionally, many correct analysis (according to reference annotations) are still not found because the rule component lacks certain types of rules especially relevant for higher-level segment types. In the following, we give an overview of the major error types that we identified through an analysis of RST annotations generated by the SemDok parser. On their basis, we point out future enhancements that would further increase its performance.

**Disambiguation of discourse markers.** The sample corpus consisting of 564 RST subtree annotations together with their indicating discourse markers is actually too small to reliably disambiguate the 96 readings of discourse markers accounted for in the discourse marker lexicon. A bigger corpus with annotations of both rhetorical relations and their indicating discourse markers thus should be acquired.

**Cues for higher segment levels.** Although the SemDok parser was projected for discourse parsing of text types with a more complex structure, so far only cues from the logical document structure (and partly anaphoric structure) are implemented to analyse rhetorical structures on higher levels than the block level. Particularly, analyses of cues from lexical chaining analyses and text type structure analyses have been prepared for inclusion [4] but not been implemented yet.



**Fig. 3.6** Reference annotation above and output of the SemDok parser.

**Syntactic and morphological annotation.** The morphological and syntactic annotations of the SemDok corpus were produced using the commercial software *Machine Syntax* from Connexor Oy. For the fairly complex sentences of the scientific articles in the SemDok corpus, however, the software frequently yields false analyses primarily because much of the domain-specific vocabulary occurring in the corpus is apparently not included in the Machine lexicon. Furthermore, the performance of discourse parsing on sentence level is negatively affected by the frequently missing or erroneous identifications of the embedding structure of paratactic sentences.

**Identification of higher-level discourse segments and scope of discourse markers.** A top-down identification of higher-level discourse segments other than those predicted by the logical document structure is currently lacking in the SemDok parser. Presently, all RST subtrees constructed during bottom-up parsing yield new complex discourse segments. This leads to an overgeneration of chart edges that currently cannot be disambiguated adequately by the scoring routine. Instead, a top-down prediction of further higher-level, complex segments would be desirable and would also help identify the *scope* of discourse markers more efficiently. Figure 3.6 shows that in the reference annotation (the above structure), the adverb *daher* (“hence”) led to a connection of the nucleus segment 4 with a satellite consisting of segments 2-3 by the CAUSE relation. In contrast, the SemDok parser identified segment 1-3 as the best-rated satellite of the CAUSE relation indicated by the discourse marker in segment 4. Apart from the fact that such an analysis seems to express an alternative but maybe also correct interpretation of segments 1-4, an independent identification of only 1-3 as a complex discourse segment could have

avoided this failed match. For this purpose, an initial thematic segmentation of an input document should be deployed, e.g. according to the algorithms of described in [15] or [13].

## References

- [1] Allen, J.: *Natural Language Understanding*, 2nd edn. Benjamin/Cummings, Redwood City (1994)
- [2] Asher, N., Lascarides, A.: *Logics of Conversation*. Cambridge University Press, Cambridge (2003)
- [3] Asher, N., Vieu, L.: Subordinating and coordinating discourse relations. *Lingua* 115(4), 591–610 (2005)
- [4] Bärenfänger, M., Hilbert, M., Lobin, H., Lungen, H., Puskàs, C.: Cues and constraints for the relational discourse analysis of complex text types - the role of logical and generic document structure. In: Sidner, C., Harpur, J., Benz, A., Kühnlein, P. (eds.) *Proceedings of the Workshop on Constraints in Discourse*, National University of Ireland, Maynooth, Ireland, pp. 27–34. (2006)
- [5] Bärenfänger, M., Goecke, D., Hilbert, M., Lungen, H., Stührenberg, M.: Anaphora as an indicator of elaboration: A corpus study. *JLCL - Journal for Language Technology and Computational Linguistics*, 49–72 (2008)
- [6] Bärenfänger, M., Lobin, H., Lungen, H., Hilbert, M.: OWL ontologies as a resource for discourse parsing. *LDV-Forum GLDV-Journal for Computational Linguistics and Language Technology* 23(2), 17–26 (2008)
- [7] Carlson, L., Marcu, D., Okurowski, M.E.: *RST discourse treebank* (2002), <http://www ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07> (visited 20.01.2009), Linguistic Data Consortium
- [8] Corston-Oliver, S.H.: Identifying the linguistic correlates of rhetorical relations. In: *Proceedings of the ACL Workshop on Discourse Relations and Discourse Markers*, pp. 8–14 (1998)
- [9] Cramer, I., Finthammer, M.: An evaluation procedure for word net based lexical chaining: Methods and issues. In: *Proceedings of the Global WordNet Conference 2008*, Szeged, Hungary (2008)
- [10] Diewald, N., Stührenberg, M., Garbar, A., Goecke, D.: Serengeti – Webbasierte Annotation semantischer Relationen. *JLCL - Journal for Language Technology and Computational Linguistics*, 74–94 (2008)
- [11] Earley, J.: An efficient context-free parsing algorithm. *Communications of the Association for Computing Machinery* 13(2), 94–102 (1970)
- [12] Egg, M., Redeker, G.: Underspecified discourse representation. In: Benz, A., Kühnlein, P. (eds.) *Constraints in Discourse, Pragmatics & Beyond*, Benjamins, Amsterdam, pp. 117–138 (2008)
- [13] Green, S.J.: Lexical semantics and automatic hypertext construction. *ACM Computing Surveys* 31(4) (1999)
- [14] Hanneforth, T., Heintze, S., Stede, M.: Rhetorical parsing with underspecification and forests. In: *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*, Edmonton, Canada (2003)

- [15] Hearst, M.A.: TextTiling: A quantitative approach to discourse segmentation. Technical Report UCB:S2K-93-24 (1993), <http://people.ischool.berkeley.edu/hearst/tiling-about.html> (visited 20.01.2009)
- [16] Hellwig, P.: Parsing natürlicher Sprachen: Grundlagen und Parsing natürlicher Sprachen: Realisierungen. In: Bátori, I.S., Lenders, W., Putschke, W. (eds.) Computational Linguistics. An International Handbook on Computer Oriented Language Research and Applications, Handbücher zur Sprach- und Kommunikationswissenschaft, de Gruyter, Berlin, pp. 348–431 (1989)
- [17] Hilbert, M., Lungen, H.: RST-HP - Annotation of rhetorical structures in SemDok. Interne Reports der DFG-Forschergruppe 437 "Texttechnologische Informationsmodellierung", Justus-Liebig-Universität Gießen, Fachgebiet ASCL (2009)
- [18] Hilbert, M., Lungen, H., Bärenfänger, M., Lobin, H.: Demonstration des SemDok-Textparsers. In: Storrer, A., Geyken, A., Siebert, A., Würzner, K.M. (eds.) Proceedings of the 9th Conference on Natural Language Processing (KONVENS 2008), pp. 22–28. Ergänzungsband Textressourcen und lexikalisches Wissen, Berlin (2008)
- [19] Le Thanh, H.: An approach in automatically generating discourse structure of text. Journal of Computer Science and Cybernetics, Vietnam 23(3), 212–230 (2007)
- [20] Le Thanh, H., Abeyasinghe, G.: A study to improve the efficiency of a discourse parsing system. In: Gelbukh, A. (ed.) CICLE 2003. LNCS, vol. 2588, pp. 104–117. Springer, Heidelberg (2003)
- [21] Le Thanh, H., Abeyasinghe, G., Huyck, C.: Using cohesive devices to recognize rhetorical relations in text. In: Proceedings of the 4th Computational Linguistics UK Research Colloquium (CLUK-4). University of Edinburgh, UK (2003)
- [22] Le Thanh, H., Abeyasinghe, G., Huyck, C.: Automated discourse segmentation by syntactic information and cue phrases. In: Proceedings of the IASTED International Conference on Artificial Intelligence and Applications (AIA 2004), Innsbruck, Austria (2004)
- [23] Le Thanh, H., Abeyasinghe, G., Huyck, C.: Generating discourse structures for written texts. In: Proceedings of COLING 2004, Geneva, Switzerland (2004)
- [24] Lenz, E.A., Lungen, H.: Dokumentation der Annotationsschicht: Logische Dokumentstruktur. Internal Report, Universität Dortmund, Institut für deutsche Sprache und Literatur/ Justus-Liebig-Universität Gießen, Fachgebiet ASCL (2004), <http://www.uni-dortmund.de/hytext/hytext/publikationen.html>
- [25] Lungen, H., Puskás, C., Bärenfänger, M., Hilbert, M., Lobin, H.: Discourse segmentation of german written texts. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) FinTAL 2006. LNCS (LNAI), vol. 4139, pp. 245–256. Springer, Heidelberg (2006)
- [26] Lungen, H., Kunze, C., Lemnitzer, L., Storrer, A.: Towards an integrated OWL model for domain-specific and general language wordnets. In: Proceedings of the Fourth Global WordNet Conference (GWC 2008), Szeged, Hungary, pp. 281–296 (2008)
- [27] Lungen, H., Bärenfänger, M., Hilbert, M., Lobin, H., Puskás, C.: Discourse relations and document structure. In: Metzger, D., Witt, A. (eds.) Linguistic Modeling of Information and Markup Languages. Contributions to Language Technology, Text, Speech and Language Technology. Springer, Dordrecht (2010)
- [28] Magerman, D.M., Marcus, M.P.: Pearl: A probabilistic chart parser. In: Proceedings of the European ACL Conference, pp. 40–47 (1991)

- [29] Mann, W.C., Thompson, S.A.: Rhetorical Structure Theory: Toward a functional theory of text organisation. *Text* 8(3), 243–281 (1988)
- [30] Marcu, D.: The rhetorical parsing, summarization, and generation of natural language texts. PhD thesis, University of Toronto (1997)
- [31] Marcu, D.: *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge (2000)
- [32] Naumann, S., Langer, H.: *Parsing*. Teubner, Stuttgart (1994)
- [33] Polanyi, L., Culy, C., van den Berg, M., Thione, G.L., Ahn, D.: A rule based approach to discourse parsing. In: *Proceedings of the 5th Workshop in Discourse and Dialogue*, Cambridge, MA, pp. 108–117 (2004)
- [34] Polanyi, L., Culy, C., van den Berg, M., Thione, G.L., Ahn, D.: Sentential structure and discourse parsing. In: *Proceedings of the ACL 2004 Workshop on Discourse Annotation*, Barcelona, pp. 49–56 (2004)
- [35] Reitter, D.: Rhetorical analysis with rich-feature support vector models. Master's thesis, University of Potsdam (2003)
- [36] Reitter, D.: Simple signals for complex rhetorics: On rhetorical analysis with rich-feature support vector models. In: Seewald-Heeg, U.: (ed) *Sprachtechnologie für die multilinguale Kommunikation. Textproduktion, Recherche, Übersetzung, Lokalisierung. Beiträge der GLDV-Frühjahrstagung, Köthen, LDV-Forum*, vol. 18(1,2), pp. 38–52 (2003)
- [37] Reitter, D., Stede, M.: Step by step: Underspecified markup in incremental rhetorical analysis. In: *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC 2003) at the EACL, Budapest* (2003)
- [38] Soricut, R., Marcu, D.: Sentence level discourse parsing using syntactic and lexical information. In: *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference (HLT/NAACL)*, Edmonton, Canada (2003)
- [39] Tapanainen, P., Järvinen, T.: A non-projective dependency parser. In: *Proceedings of the 5th Conference on Applied Natural Language Processing, Association for Computational Linguistics, Washington D.C.*, pp. 64–71 (1997)
- [40] Tomita, M.: An efficient augmented-context-free parsing algorithm. *Computational Linguistics* 13(1-2), 31–46 (1987)
- [41] Walsh, N., Muellner, L.: *DocBook: The Definitive Guide*. O'Reilly, Sebastopol (1999)
- [42] Hilbert, M., Lungen, H., Bärenfänger, M., Lobin, H.: Demonstration des SemDok-Textparsers. In: Storrer, A., Geyken, A., Siebert, A., Würzner, K.M. (eds.) *Proceedings of the 9th Conference on Natural Language Processing (KONVENS 2008)*, pp. 22–28. *Ergänzungsband Textressourcen und lexikalisches Wissen*, Berlin (2008)
- [43] Webber, B.: D-LTAG: Extending Lexicalized TAG to Discourse. *Cognitive Science* 28(5), 751–779 (2004)
- [44] Witt, A.: Multiple hierarchies: New aspects of an old solution. In: *Proceedings of the Extreme Markup Languages, Montreal* (2004)
- [45] Witt, A., Lungen, H., Goecke, D., Sasaki, F.: Unification of XML documents with concurrent markup. *Literary and Linguistic Computing* 20(1), 103–116 (2005)