Uladzimir Sidarenka, Andreas Peldszus, Manfred Stede

Discourse Segmentation of German Texts

This paper addresses the problem of segmenting German texts into minimal discourse units, as they are needed, for example, in RST-based discourse parsing. We discuss relevant variants of the problem, introduce the design of our annotation guidelines, and provide the results of an extensive inter-annotator agreement study of the corpus. Afterwards, we report on our experiments with three automatic classifiers that rely on the output of state-of-the-art parsers and use different amounts and kinds of syntactic knowledge: constituent parsing versus dependency parsing; tree-structure classification versus sequence labeling. Finally, we compare our approaches with the recent discourse segmentation methods proposed for English.

1 Introduction

'Discourse parsing' nowadays typically refers to the task of assigning a structure to a monologue text, where this structure is driven by an underlying theory of coherence relations and their composition. One popular such theory is Rhetorical Structure Theory (RST, Mann and Thompson (1988)), which holds that a tree can be obtained by recursively relating adjacent 'elementary discourse units' (EDUs). Early work on deriving RST trees automatically was done by Marcu (2000), and following the advent of a large data set, the RST Discourse Treebank (Carlson and Marcu, 2001), a variety of machine learning approaches have been proposed to tackle the problem, for example those of Hernault et al. (2010b) and Ji and Eisenstein (2014).

An alternative theory is Segmented Discourse Representation Theory (SDRT, Asher and Lascarides (2003)) which does not enforce the tree-structure constraint and is also applicable to dialogue. Corpora are available in English and French, and one of the SDRT parsers that have been presented recently is the one by Muller et al. (2012).

For RST and SDRT (and similar approaches), any discourse parser relies on a segmentation of the text into EDUs, which in all present frameworks amounts to a sequence of non-overlapping units that completely covers the text. While the original RST paper by Mann and Thompson remained relatively vague on the issue of defining EDUs (the essential characterization was "typically clauses"), work on discourse parsing relies on an operationalization and thus on a specific definition. We will discuss the issues involved in this step below in Section 2.

RST and SDRT are in contrast with the Penn Discourse Treebank approach to discourse structure (Prasad et al., 2008), which does not assign any complete structure to the text, but marks up individual coherence relations (which may or may not be explicitly signaled by a connective or other lexical means) and their argument spans.

Each relation is annotated individually, without considering any surrounding structure. And since annotators do not receive specific instructions on what may constitute an argument of a connective, there is no need for a definition of EDU in this particular approach. However, it is empirically interesting to compare the arguments that were selected intuitively by the annotators to a formal notion of EDU and determine where they match and where they do not.

Discourse parsing is not the only application that needs EDU segmentation. Work on speech act assignment, which originated in the dialogue community but has spread to the annotation of social media contributions and sometimes also to "standard" monologue texts (e.g., Stede and Peldszus (2012)) also relies on the notion of a minimal unit of text that can be ascribed a speech act. One extension of this is the increasingly popular area of argumentation mining, where argumentative moves and relations among them are being identified. Another research field which also might significantly benefit from a high-quality discourse segmentation is that of anaphora resolution. For any implementation of Centering Theory (Grosz et al., 1995), a discourse segmentation is the prerequisite for computing centering transitions, which in turn influence the assignment of pronoun antecedents. In an interesting study, Taboada and Zabala (2008) demonstrated the effects of different EDU definitions on pronoun resolution performance. Likewise, approaches in the "constraints and preferences" tradition (Lappin and Leass, 1994), which compute salience rankings over sequences of minimal discourse units, rely on a definition of EDU.

While a lot of work has been done along the lines described above for English, very few studies have been presented on German, and we will mention them later in the paper. With this paper, we hope to close this gap by detailing our guidelines for the manual annotation of discourse segments in a German corpus (Potsdam Commentary Corpus, Stede and Neumann (2014)), conducting an extensive inter-annotator agreement study of the resulting annotation, and offering the first generally-available discourse segmentation module that was specifically designed and implemented for German.

The rest of this work is structured as follows. Section 2 introduces the problem of discourse segmentation in more depth and discusses different options for defining the notion of EDU. Then, in Section 3, we describe our annotation guidelines and present the results of an annotator agreement study as well as a brief description of the resulting annotated corpus. After summarizing related work on discourse segmentation for English and German in Section 4, we turn our attention to automatic segmentation methods and describe three different approaches involving classifiers that operate on the output of state-of-the-art German syntax parsers. In particular, our interest is in comparing the performances of a dependency parser and a constituency-based parser for our particular task; this is then supplemented by the analysis of a sequence labeling approach. The performance of the three methods is evaluated on our manually-annotated corpus. Finally, Section 5 discusses our results and draws conclusions.

2 Discourse Segmentation: The Problem

As Mosegaard-Hanse (1998) observed, the task of segmentation can in principle be performed on the basis of

- *form*, by providing structural criteria for boundary identification; or
- *meaning*, by identifying stretches of text that express complete propositions, and assigning segment boundaries accordingly; or
- *action*, by identifying stretches of text that represent complete speech acts, and assigning their boundaries.

From a computational perspective, the first option is the "ideal" one: If it were possible to exploit formal signals only, any task involving language unit *interpretation* could be clearly split off from a preparatory step of purely form-based unit *identification*. This would certainly work if human language users communicated solely in terms of sequences of main clauses. But obviously matters are more complicated, mainly because

- complex sentences for many purposes need to be split into individual clauses, and
- fragmentary material of various kinds needs to be accounted for.

In some approaches, the problem is "solved" by generally taking the complete sentence, including all minor clauses and adverbials, as unit of analysis. In local-coherence analyses based on Centering, this was done regularly (Tetreault, 2001; Miltsakaki, 2002). But this was hardly a decision on the grounds of theoretical adequacy or empirical evidence, but on the grounds of practical convenience. Taboada and Zabala (2008) discussed this in detail and proposed a more fine-grained segmentation for the application of Centering.

Similarly, when the goal is to annotate coherence relations, as with RST, defining smaller units is inevitable, since such relations often hold intra-sententially. Again, the importance of high-quality segmentation can be demonstrated by its effect on the task, here on RST parsing: Soricut and Marcu (2003) in their study determined that the availability of perfect (gold standard) segmentation would reduce the number of discourse parser errors by 29%. Therefore, finding good solutions to the EDU demarcation task is of great importance.

Nonetheless, in empirical analyses, it proved rather difficult to make boundary decisions solely based on formal criteria. For example, Marcu (2000) in his pioneering work gave a few structural criteria for RST segmentation, but then added that further boundaries are to be introduced "when a coherence relation is present", thus effectively combining 'form' and 'meaning' criteria. As will become clear in the next section, in our project we also in general follow this approach, but the meaning-aspect of boundary assignment is accounted for by a more general "interpretative" criterion.

Once the basic decision is made to break sentences into smaller units, the target level of granularity needs to be determined. One extreme position is advocated by Schmitt

(2000), who, for the purposes of illocution analysis, accepts some individual adverbs as complete units, since they can express a separate illocution (most often an evaluation) supplementing that of the clause. The vast majority of approaches, however, adopts a much less radical position and takes the presence of a verb as a central condition. We also follow this line in our own approach and take the clause as a basis for the analysis, with exceptions made for fragmentary material that occurs with sentence-final punctuation (see the next section).

Finally, when defining a segmentation task, it needs to be decided whether the output should be a 'flat' sequence of units (of equal status), or whether it should mirror syntactic structure to some extent. Again, this of course depends on the purpose: A *topic-based* segmentation of a text, e.g. in the 'text tiling' tradition (Hearst, 1997), is flat in the vast majority of approaches. Also, the EDUs in RST parsing, in the end, constitute a flat partitioning, but here, the segmentation step can benefit from hierarchical information when determining embedded EDUs. Paying attention to embedding is also of great importance in illocution-oriented analyses, where independent speech acts need to be identified. Embedding can be represented to different degrees and in different ways, viz., by explicitly providing the bracketing structure or by adding syntactic type labels to EDUs that otherwise would form a flat sequence.

For our corpus, we want to be open to various tasks and chose to annotate clause hierarchy (i.e., to provide a very coarse-grained syntactic analysis) and to label the units with their structural types. The different annotation tasks that build upon the segmentation can then either peruse or ignore the structural embedding and the labels.

3 Human Annotation Study

The data for our study comes from the Potsdam Commentary Corpus (PCC, Stede and Neumann (2014)), a freely-available collection of 176 newspaper editorials from a German regional newspaper. The PCC is being distributed with various layers of manually-produced annotations: sentence syntax, nominal coreference, connectives and their arguments, and rhetorical structure. The work reported in the present paper adds a new layer of discourse segments to the corpus. To this end, we devised annotation guidelines and conducted an annotator agreement study, which we present in this section.

3.1 Annotation guidelines

The idea behind our guidelines (Stede et al., 2015) is to provide a base segmentation that can be utilized by other layers of text annotation, such as analyses of illocutionary force, rhetorical structure, coreference, or argumentation. These different purposes can make use of a segmentation in slightly different ways; therefore we aim at a layer of EDUs that is relatively fine-grained, provides type information for the units, and can be thus systematically mapped to reduced, more coarse-grained versions.

A central design decision was that our guidelines for manual annotation take both structural and subjective-interpretation features into consideration. Annotators are asked to identify complete 'sense units', chunks of information that convey a sense of completeness. This clearly subjective criterion is intertwined with the guideline to have most decisions revolve around sentence-final punctuation symbols (full stop, colon, exclamation mark, question mark; henceforth: SFPS). And besides finding boundaries, the annotators are asked to assign a syntactic type label to each unit.

Specifically, our annotators proceed in three phases, with each one being applied to the complete text:

1. For each SFPS, check whether it finishes off a complete sense unit; i.e., make sure that the current sense unit does not stretch beyond this SFPS. If the sense unit does stop at the SFPS, mark it as a sense unit boundary.

2. For each sense unit, check whether it contains more than one structural unit, i.e., one that ends with a SFPS. This can be a full sentence or a fragment; assign appropriate syntactic type labels to each such unit.

3. For each full sentence, check whether it is structurally complex, i.e., it contains several clauses or parenthetical material. If it does, provide markup for the structure.

The result of the procedure is a tree-like structure spanning the complete text: a sequence of contiguous sense units, each of which may consist of recursively embedded structural units. In the following, we provide some details about the three phases of annotation and illustrate them with examples.

3.1.1 Step 1: Identify sense units

The idea of this step is to break the text into interpretable units. We constrain the possible positions of unit boundaries: They can occur only at the punctuation symbols. This largely leads to standard sentence segmentation, but it also takes care of possible fragmentary material that does not correspond to a full sense unit but is to be amalgamated with the preceding or subsequent sentence. Our criterion of "complete sense unit" is to be tested after removing any connectives and after (mentally) replacing anaphoric material with their antecedents.

Here are a few examples of material that contains sentence-final punctuation but is to be fused with a neighbor unit in order to form a complete sense unit:

(1) Most important is: Always keep your eyes open.

(2) The boy bought himself an ice cream. And another one.

(3) There was only one thing that could save me. A good book.

3.1.2 Step 2: Subdivide sense units

As the examples given above illustrate, fragments can be attached either to their left neighbor (because they provide an extension) or to their right neighbor (because they introduce it). One purpose of step 2 is to make this decision explicit by assigning them different types (initiating fragment, FRE; versus finalizing fragment, FRB).[1]

The other purpose is to give types to the non-fragments, i.e., the "ordinary" material. For the most part, this will be main clauses, which are marked as 'HS' *(Hauptsatz)*. However, we distinguish the full main clauses from reduced ones, where the 'fragment' is not to be adjoined with one of the neighbors. These incomplete main clauses (HSF, *Hauptsatzfragment*) are assigned when the clause is elliptical (but can be easily filled from the preceding context) or when it constitutes a complete illocution.

(4) [*Most important is:*]$_{HSF}$ [*Always keep your eyes open.*]$_{HS}$

(5) [*I bought a new laptop.*]$_{HS}$ [*And a camera.*]$_{HSF}$

Thus at the end of step 2, the text is broken into a sequence of labeled units: (two types of) fragments, main clauses, and incomplete main clauses.

3.1.3 Step 3: Subdivide complex sentences

This step is in charge of recursive embedding, where three cases are to be distinguished.

Parataxis: main clauses that appear in the same sentence, possibly linked by a coordinating conjunction. Each receives the label *HS*.

Hypotaxis: Largely following the inventory presented by Bußmann (2002), we distinguish nine different kinds of minor clause, among them subject clause, object clause, adverbial clause, predicative clause, relative clause. Annotators need to determine the clause type and assign the appropriate label. Minor clauses can also be conjoined, in which case we mark them individually.

Parentheticals are units that "interrupt" the clause and are marked by commas, hyphens, or parentheses. However, we mark only those that correspond to a complete proposition or a clearly identifiable illocution.

The following examples[2] illustrate our usage of categories for different types of embedded units.

(6) [*Gestern hat der Lehrer* [*– ganz schön lächerlich! –*]$_{HSF}$ *mit blauen Briefen für verspätete Schüler gedroht.*]$_{HS}$
('Yesterday the teacher – quite ridiculously! – threatened late-coming pupils with sending letters to their parents.')

(7) [*Heute war,* [*so soll es ja auch sein,*]$_{HS}$ *das Kind pünktlich in der Schule.*]$_{HS}$
('Today, as it should be, the child arrived at school on time.')

[1] The abbreviations of all our types are compiled in Table 2 below.
[2] Since the categories do not straightforwardly map to English, we give German examples here.

(8) [*Heute war das Kind* [*(das öfters mal Probleme mit dem Aufstehen hat)*]$_{ANR}$ *pünktlich in der Schule.*]$_{HS}$
('Today the child (who often has problems with getting out of bed) arrived at school on time.')

3.2 Agreement study

From our corpus, we selected 10 texts, each of which is approximately 180 words or 1100 characters long. Two annotators (one Ph.D. student and one post-doc, both with linguistic background) annotated these texts separately, after studying the 15-page guideline document.

3.2.1 Methodological issues

Choosing the right metric for evaluating the results of our agreement study is not trivial. Three desiderata are important for an ideal metric: The metric should be chance-corrected to compensate for expected chance agreement; furthermore, it should be appropriate for the annotation task, i.e., be able to account for all aspects of the annotated structure; finally, it should be well understood, so that there is a consensus on how to interpret the results and what constitutes "good" agreement.

Flat segmentation metrics: There exist different metrics to evaluate flat text segmentation (as for example for the task of topic segmentation, argumentative zoning, or discourse segmentation). Prominent ones are P_k (Beeferman et al., 1999) and WinDiff (Pevzner and Hearst, 2002). Both metrics measure the boundary agreement by moving a window of fixed size over two segmentations of the same sequence and checking whether both agree that the window's edges are in the same segment or not. WinDiff was proposed to overcome insensitivities for certain error types that P_k exhibited. However, both metrics are not chance-corrected. Therefore, to access the reliability of segmentation annotation more accurately, Krippendorff (1995) presented α_U, an agreement measure for unitizing in the family of alpha coefficients. An alternative, a Fleiss multi-π agreement coefficient based on boundary edit distance π^*_{BED} has been presented by Fournier (2013).

Even though our annotations are hierarchical, we will present results in these metrics in order to allow for comparisons with past segmentation studies. For this purpose, we flattened the annotated tree structures to a fine-grained partitioning, with a boundary inserted at every constituent border: *(a (b (c d)))* $\mapsto$ | *a* | *b* | *c d* | . It is to be noted, however, that flat segmentation will only be able to represent embeddings above a depth of 1 when the units are at the left or right border of the superordinate segment, because discontinuous segments (as they would result from center embedding) cannot be captured.

Category agreement metric: All of the above metrics are untyped, i.e., they capture the distinction between boundary or no boundary, but do not consider segments of different categories. If the extensions of spans of two annotators match, agreement metrics for categorical data such as Cohen's κ (Cohen, 1960) can be used to assess

the agreement of category assignment. With this metric, the category assignment for embedded structures can be evaluated without the need of flattening. However, whenever segmentations are different, this simple form of categorial agreement cannot systematically be applied. A generalization of α_U for segments with different categories has been given by Krippendorff (2004). It is not only able to assess the agreement in segmentation but also in segment categorization. In this paper, we will use the symbol $\alpha_{\mathrm{U}}^{\mathrm{c}}$ for this metric.[3]

For using the $\alpha_{\mathrm{U}}^{\mathrm{c}}$ metric on our segmentation trees, we again have to flatten the trees, this time with category labels: *(X a (Y b (Z c) d) e)* $\mapsto$ $|_X\ a\ |_Y\ b\ |_Z\ c\ |_Y\ d\ |_X\ e\ |$. As before, the mapping splits nodes with center embedding into an opening, an embedded, and a closing segment. The opening and the closing segment will be of the same type.

Tree metrics: More appropriate for the comparison of our annotations are tree metrics. In parser evaluation, phrase-structure trees are typically compared with the parseval metric (Black et al., 1991), i.e., labeled and unlabeled precision, recall, and $F1$. The unlabeled scores will reflect the structural agreement of segmentation, the labeled ones will furthermore reflect the category assignment. However, parseval is also not directly suited for accessing inter-annotator agreement, first because it assumes one representation to be the correct one, and second because it is not chance-corrected.

3.2.2 Results

The results of the agreement study are presented in Table 1. The first four rows represent scores for flat segmentation: the uncorrected scores metrics P_k and WinDiff, and the corrected metrics α_{U} and π^*_{BED}. Note that P_k and WinDiff are error measures, where small numbers are desired. The next rows present $\alpha_{\mathrm{U}}^{\mathrm{c}}$ for typed, flat segmentations and unlabeled and labeled $F1$ scores. The last two rows of the table represent the ratio of exact boundary matches and the κ agreement on the categories of those matches. Except for these two rows, all other metrics are reported as average percentages with standard deviation over the evaluated texts. We also performed similar calculations on the whole corpus, but this yielded very similar scores.

The error in flat segmentation measured with P_k and WinDiff is remarkably low. According to these metrics, seven of ten texts have a perfect agreement, and only one text stands out with an error rate of 11-12%. The chance-corrected agreement measured with π^*_{BED} and α_{U} is consequently very high. However, note that the flat segmentation neither accounts for the actual embedding nor for the correctness of the segments' types. The $\alpha_{\mathrm{U}}^{\mathrm{c}}$ metric takes these categories into account. It is on average on a very high level. Most texts yield nearly perfect results, only three texts fall out with agreement scores around 60. For the tree metrics, unlabeled and labeled F-score are both around 90%. Three of the texts reach perfect agreement, another three attain nearly perfect F-scores of 95-99%. For the labels, there were only a few disagreements about the categories of subordinate clauses. Here, the most frequent confusion was between

[3] For the calculation of all coefficients of the alpha family, we use the implementation of Meyer et al. (2014).

Metric	Anno_1 vs Anno_2	
	Mean	Std.Dev.
P_k	01.56	±3.55
WinDiff	01.77	±4.04
π^*_{BED}	96.92	±6.64
α_{U}	98.46	±4.07
α^c_{U}	85.95	±17.03
parseval unlab. $F1$	90.18	±7.72
parseval lab. $F1$	89.13	±7.75
matching spans	89.91	
categories κ	88.56	

Table 1: Results of the agreement study between the two annotators in different metrics. For details about the metrics, see Section 3.2.1.

subject and object clauses, typically occurring in the context of expletive constructions, where annotators found it hard to correctly identify the grammatical role of the minor sentence. Structural differences are minimal and mostly occur when one annotator decides for a more fine-grained sub-clausal segmentation. The strongest drop in terms of F-score was observed when one annotator repeatedly split up a conjunction of main sentences, which are supposed to be enclosed by one large main sentence node, into independent main sentences without an enclosing sentence node. The high structural agreement is also reflected by the high ratio of matching spans: Nearly 90% of the spans were exact matches and yielded a substantial agreement of 0.88 κ for segment categories.

It is worth pointing out that the results in Table 1 do not consider the level of sense units (which is step 1 of the annotation procedure, see Section 3.1.1). Identifying sense units is a task that requires a deeper understanding of the text, something that is much easier to achieve for human annotators than for computational models of text processing. We decided to exclude nodes of the sense unit level, in order to facilitate the comparison with the automatic segmenters that we will present in Section 4. Nevertheless, we want to report the annotator agreement for the full annotation task including sense unit identification: All flat segmentation metrics are unaffected by an additional level of nodes in the segmentation trees and thus yield equal results. Only the tree metric reflects the increased structural complexity: The annotators achieve an unlabeled $F1$-score of 89.72 ±11.88, and a labeled $F1$-score of 88.58 ±11.53. These figures are on average only slightly lower than those presented in Table 1, but show a higher variation.

To sum up, the agreement between the annotators is very high, allowing us to conclude that discourse segmentation can be reliably annotated with our scheme. In the light of the above discussion, it is not straightforward to compare the results with others given in the literature, but we want to mention that Tofiloski et al. (2009) measured the annotator agreement for their flat, untyped segmentation task on 10 English texts, and reported a κ of 0.85. For German, the only result we are aware of is the experiment by Versley and Gastel (2012), which lead to a $\kappa > 0.9$, likewise for flat, untyped segmentation.

3.3 Corpus

Having obtained the promising agreement results, we proceeded to annotate the full PCC data set (mentioned at the beginning of this section). The annotation was done by a trained annotator using the EXMARaLDA annotation tool (Schmidt et al., 2011) and corrected in a later consolidation phase. The full corpus contains 176 texts, with 2,180 sentences and about 32,000 tokens. An overview of the frequency of the different segment types that resulted from the annotation is given in Table 2. For illustration, we show an original sentence from the corpus with its annotation:

(9) [*Zu einer Zeit ,* [*in der alles Denkbare auch machbar erscheint ,*]$_{ARR}$ *ist es beruhigend* [*zu wissen ,* [*dass die Rettungskräfte sich nicht erst seit gestern damit befassen ,* [*wie sie die Bürger vor Katastrophen schützen können .*]$_{OBJ}$]$_{OBJ}$]$_{SUB}$]$_{HS}$

Segment Type			Symbol	Count
Main Sentence	complete		HS	2133
	fragment		HSF	285
Minor Sentence	clause constituent	subject clause	SUB	222
		object clause	OBJ	281
		adverbial clause	ADV	346
		predicative clause	PRD	28
	attributive clause	restrictive relative clause	ARR	209
		non-restrictive relative clause	ANR	51
		participle construct	APK	8
		other	ATT	74
		expansive minor clause	WEI	8
	unclear		UNS	5
Fragment	sentence-initial		FRE	55
	sentence-final		FRB	36
				3742

Table 2: Segment types annotated in the corpus

We also analyzed the resulting hierarchical structures of the annotated segments and present the results of this analysis in Table 3. The first column of this table specifies different depths of segment embeddings. These values range from one (a simple,

uncoordinated main sentence without minor clauses) to five. The second column shows the total number of segments annotated at the given depth. Finally, the number of texts exhibiting this maximal segment nestedness is given in the third column. Notice that a clear majority of texts has a maximum embedding depth of three.

For comparison, Afantenos et al. (2010), who work with a French corpus annotated in accordance with SDRT, report that almost 10% of EDUs in their corpus are part of an embedded structure.

Depth of Embedding	Segments	Texts
1	2180	2
2	1335	63
3	206	93
4	20	17
5	1	1
total	3742	176

Table 3: Depth of embedding: The number of segments annotated at this depth (second column) and the number of texts with this maximum depth (third column).

4 Automatic Segmentation

A natural question that arises after measuring human agreement and annotating the complete corpus is how well automatic methods can perform as compared to simple baseline techniques and to the human level of expertise. In this context, we first need to know whether nested or flat segmentation would be more amenable to the automatic processing, and how much the results of the two approaches would differ. To this end, for predicting nested segments, we also have to look into what kind of syntactic information and which form of syntactic structure (syntactic constituents or dependency trees) are suited to make correct predictions about the scope and embedding level of discourse units.

We try to address these and other questions in this section. After summarizing related work on the automatic discourse segmentation of English and German,[4] we establish a straightforward comparison baseline, in which we consider every sentence as a single atomic discourse unit. We use this baseline to compare two more advanced segmentation methods that recursively apply automatic classifiers in order to predict which syntactic constituents or dependents initiate discourse segments. In the final step, we present the results of the flat state-of-the-art segmenter of Feng and Hirst (2014), which we adjusted to the peculiarities of the German language and applied to our corpus. To ease the comparison, we test all three classifiers on our original dataset but

[4] More comprehensive summaries can be found in Stede (2011, pp. 87-97) and Webber et al. (2012, pp. 448-455).

do not make a distinction between the boundaries of sense units (step 1 in the human annotation procedure) and the boundaries of other segments. Finally, we summarize our results and draw conclusions in Section 5, which also includes some suggestions for future research.

4.1 Related Work

As noted by Grosz and Sidner (1986), discourse segments serve as fundamental building blocks in virtually every discourse theory. Even if these theories might disagree on the mechanisms and final results of assembling separate segments into bigger structures, the mere necessity of defining and automatically detecting elementary discourse units has hardly ever been questioned. Accordingly, discourse segmentation plays a crucial role and has attracted much attention in the discourse research community, with by far the most work being done on English.

One of the the earliest attempts at discourse segmentation for RST was made by Le Thanh et al. (2004). In their primarily rule-based approach, the authors consecutively applied a cascade of processing steps: first reading input syntactic trees into a pushdown automaton, storing the non-terminal nodes of these trees on the automaton's stack, and then analyzing these non-terminals with a set of hand-crafted rules, once the system came across a constituency boundary. After applying special heuristics for disambiguating the placement of adverbs and determining the satellite/nucleus status of detected units, the last stage of this system extracted EDUs from clauses and strong cue noun phrases found in text. This system achieved an F-score of 80.35 % on a test corpus of 166 sentences.

Following this line of research, Tofiloski et al. (2009) proposed an automatic segmentation system called `SLSeg` which relied on 12 syntax-based rules and a set of lexical and part-of-speech constraints. The syntactic rules identified potential EDU boundaries between tree nodes, and the constraints removed spurious boundaries surrounding idiomatic phrases (for example, *as it stands*) or inserted new boundaries around units which could not be captured by the syntactic context only (e.g., phrases introduced by *in order to*).

One of the first supervised learning approaches to segmentation was developed by Soricut and Marcu (2003) as a part of the SPADE system. This system took lexicalized syntactic trees as input. The authors computed the probability of inserting a discourse boundary between a child and parent node by estimating the number of lexicalized child-parent pairs with an EDU boundary between them in their training corpus and dividing this count by the total number of all child-parent pairs found in the training set. This system attained an F-score of 83.1% when tested on a set of 38 journal articles.

A different technique was proposed by Sporleder and Lapata (2005). In their work, the authors considered discourse segmentation as a sequence labeling task and tried to solve it using the supervised Boosting approach (Schapire and Singer, 2000). Since this approach aimed not only at segmentation but also at determining the (RST-style) satellite/nucleus status of the detected units, the authors experimented both with

a joint and two-stage approach for solving these two tasks. For segmentation, the two-pass method performed significantly better than the joint technique and gave an improvement in F-score by $\approx 1.5\%$ over the method proposed by Soricut and Marcu (2003).

Fisher and Roark (2007), who obtained an F-score of almost 5% over the SPADE system, used a binary log-linear classifier for recognizing EDU boundaries. The authors experimented with three sets of features, including: *a)* basic finite-state, *b)* context-free, and *c)* a finite-state approximation of context-free features. The former two sets largely coincided with the features used by Sporleder and Lapata (2005), and Soricut and Marcu (2003). As a finite-state approximation, Fisher and Roark (2007) took the output of a shallow syntactic parser and partially lexicalized its chunks. Experiments showed that the best performance could be achieved by using all three sets of features together, thus supporting the claim that full syntax parsing does contribute favorably to discourse segmentation.

Finally, current state-of-the-art results for discourse segmentation of English were obtained by the two-pass system of Feng and Hirst (2014). This system also relies on a sequence labeling approach. Similarly to Soricut and Marcu (2003), the method makes its predictions over pairs of tokens rather than single words. But instead of operating on syntactic trees, this approach expects plain token sequences as input and only incorporates syntax information in the form of features associated with these token pairs. In the first stage, a supervised CRF-classifier makes initial guesses about the potential segment boundaries, which are subsequently corrected by another CRF-model. As shown by Feng and Hirst (2014), both of these strategies (predicting over token pairs and making two-pass predictions) have a crucial positive effect on the net segmentation results, achieving a F-score of 92.6% on the recognition of in-sentence boundaries.

To the best of our knowledge, the only reported attempt at discourse segmentation of German was made by Lüngen et al. (2006). In the initial guessing phase, this approach introduces a potential segment boundary for every comma, conjunction, or parenthesis found in a sentence. In the next step, a special rule-based filter removes margins surrounding enumerations, relative clauses, clausal and infinitival complements, as well as proportional clauses (*the more A, the B*), since these elements do not form independent discourse segments according to the authors' guidelines. The resulting flat segmentation was tested on a corpus of four scientific and two web-published articles, showing an average F-score of 75.57% for the recognition of in-sentence boundaries.

4.2 Baseline

In order to compare our system with these and other approaches, we establish a simple baseline to see how different techniques perform with respect to this rather simplistic method. For this purpose, we have implemented a simple segmentation module which creates a single discourse unit for each sentence identified by a customary sentence splitter (specifically, we are using the one from the OpenNLP tool suite[5]). This method

[5] `https://opennlp.apache.org/`

is expected to work correctly for most of the sentences except for the cases when the annotators identified sub-clause EDUs or considered incomplete main clauses (see the first two steps of the annotation procedure) as separate discourse units.

4.3 Hierarchical Segmentation

4.3.1 Constituency Parser Model

The first segmentation system that we are going to compare against the baseline is called `BitParSegmenter`. As suggested by the name, this system makes its predictions over syntactic constituents that are obtained from the output of the BitPar constituency parser (Schmid, 2004). In all our experiments, we used the parsing model trained on the TIGER corpus (Brants et al., 2004).

In order to train our segmentation model, we automatically processed raw sentences from our corpus with BitPar. This gave us a set of 1,911 constituency trees with a total of 50,402 non-terminal and 32,872 terminal nodes (tokens).[6] Since the results of the built-in BitPar tokenizer disagreed with the gold tokenization from our segmentation data, we next applied the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) in order to unify both splittings. In a concluding step, we consecutively aligned each constituent of every parse tree with a corresponding discourse segment (if there was one). To find such correspondences, we represented each constituent in question as a set of uniquely numbered tokens that belonged to that node, translated this token set to a respective set of discourse tokens using the previously computed alignment, and eventually checked whether there was a segment in our gold data that fully agreed with the tokens belonging to the constituency node under scrutiny.

By applying this procedure, we were able to align 2,941 out of 50,320 non-terminals (5.84%) with at least one discourse segment (we skipped those non-terminals which consisted solely of punctuation marks or whose aligned tokens resulted in an empty list for the discourse tokenization). A detailed breakdown of 10 most frequently matching constituent and segment types is given in Table 4. Conversely, 78.76% of all discourse segments had a corresponding constituent in the parse trees. This figure also gives us an upper bound on the classification results for our segmenter (i.e., even with a perfect recognition of which constituents initiated which types of segments, we still would be able to correctly reconstruct only $\approx 80\%$ of all EDUs).

After aligning syntactic constituents with their respective discourse counterparts, we constructed the training set by extracting features from every constituency (sub-)tree and taking the type of the discourse segment aligned with its top constituent as the gold label for our prediction. (Sub-)trees which did not have a corresponding discourse segment were assigned the gold class NONE.

We used the following types of attributes as features:

- the set of all terminals (tokens) comprised by the top constituent;

[6] Due to the automatic splitting with BitPar's scripts, the number of tokens and constituency trees in this set differs from the number of words and sentences in the manually labeled corpus.

Constituent Type	Segment Type	Count
TOP	HS	1,432
S-TOP	HS	253
S-MO	ADV	153
S-RC	ARR	140
TOP	HSF	100
S-OC	OBJ	86
S-SB	SUB	84
S-OC	HS	44
S-RC	ANR	40
VP-OC/zu	OBJ	39

Table 4: Most frequently matching constituent and segment types.

- the first and the last token of the head constituent as two separate features;
- the syntactic label of the head node and the syntactic label of its right-most descendant;
- the syntactic type of the parent (sub-)tree, if there was one;
- the syntactic type, the first, and the last tokens of the immediate left and right siblings of the (sub-)tree in question;
- and, finally, the height of the tree as a numeric feature.[7]

After comparing several different machine learning approaches including the random forest classifier (Liaw and Wiener, 2002), the decision tree method (Breiman et al., 1984), and the k-nearest neighbors algorithm (Fix and Hodges, 1989), we chose the linear support vector classification system (Fan et al., 2008) with the Crammer-Singer multi-class strategy (Crammer and Singer, 2002) due to both its superior performance and much faster training times as compared to the other methods.

Once the classifier was trained, obtaining the final automatic segmentation was relatively straightforward: We simply traversed each node of the input syntactic trees in the depth-first-search order and let the trained model predict the segment class of the processed nodes. Whenever the classifier made a prediction other than NONE (i.e., it decided that the constituent in question was in fact giving rise to a segment), we constructed an EDU of the predicted class for this constituent and recursively processed the children of that syntactic node, storing the results of this recursion as leaves of the newly created EDU (these results in turn could be either plain tokens or further

[7] Notice that, since BitPar does not include POS tags in its output, our features do not make use of them in this model.

discourse segments). A sample constituency tree with the classifier's predictions (shown in brackets next to the node names) and resulting segmentation is shown in Figure 1.

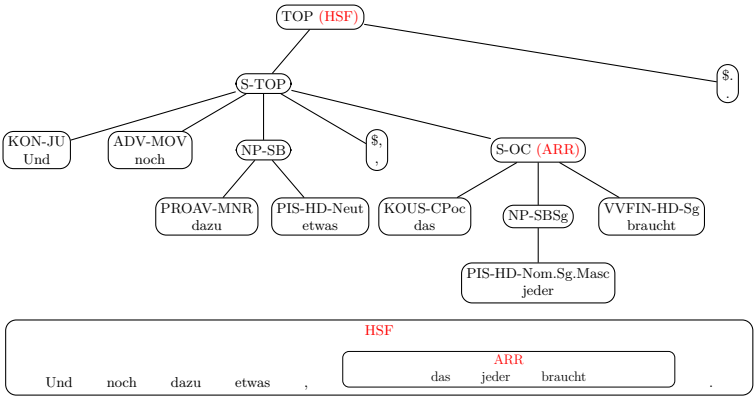


Figure 1: Example of a constituency tree with resulting discourse segmentation.

4.3.2 Dependency Parser Model

The second segmentation system that we are going to compare against the baseline is called `MateSegmenter`. This system makes its predictions over the sub-graphs of a dependency parse derived by the mate-parser (Bohnet, 2010).

In order to generate a training set for this approach, we first parsed the raw corpus with the mate-parser, getting 2,013 dependency graphs with a total of 32,838 tokens.[8] Similarly to the training procedure for the previous model, we then aligned the gold tokenization with the automatic token splitting using the Needleman-Wunsch algorithm. In the next step, all dependency sub-graphs were aligned with the annotated discourse segments by matching spans of uniquely numbered tokens. Of all dependency sub-graphs, 2,983 directly corresponded to a discourse segment (9.7% of all dependency sub-graphs). Conversely, 2,989 of the annotated discourse segments had a corresponding dependency sub-graph (79.8% of all discourse segments). This gave us a similar upper bound for the automatic classification as with the constituency parser approach.

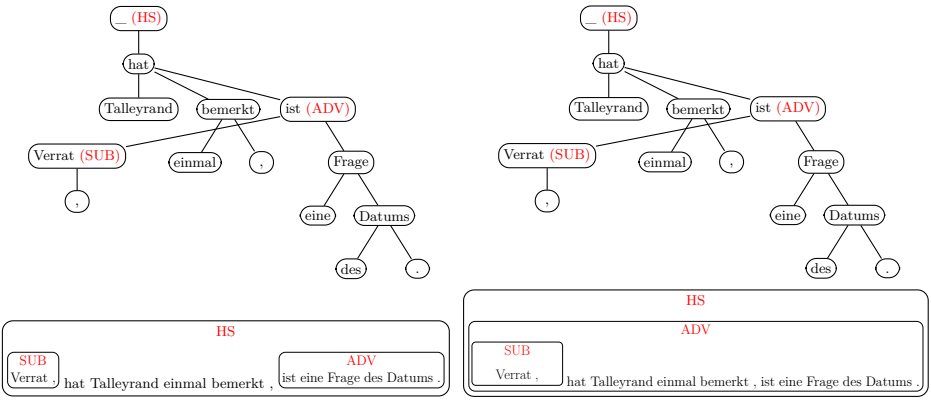
The classification items were constructed by extracting features from every dependency sub-graph. The target class for each item was the type of the aligned discourse segment or NONE, if no alignment could be established. We used the following types of attributes of the sub-graph as features:

- the token and the part-of-speech of the sub-graph's root;

[8] As with the previous parser model, the number of tokens and dependency trees differs from the number of words and sentences in the manually labeled corpus due to the automatic splitting.

- the token and the part-of-speech of the head of the sub-graph's root;
- the dependency relation between the sub-graph's root and its head;
- pairwise and triple combinations of the above three features;
- the first and the last token of the sub-graph's span;
- the token to the left and to the right of the sub-graph's span;
- unigram features for all tokens in the sub-graph's span;
- the length of the sub-graph's span measured in tokens;
- the number of punctuation tokens in the sub-graph's span.

As with the previous method, we compared several machine learning approaches and chose the linear support vector classifier. However, the construction of the final segmentation with the trained model was not as easy as for the `BitParSegmenter`. This was mainly due to the presence of non-projective edges in the input dependency trees. Once an ancestor node of such an edge was predicted to initiate a segment, simply putting all descendants of that node into a single discourse segment resulted in intertwined discontinuous discourse units, which was not a valid segment structure according to our guidelines.



(a) Greedy approach. **(b)** Generous approach.

Figure 2: Examples of dependency trees with resulting discourse segmentation.

To overcome this issue, we devised two different approaches: *greedy* and *generous*. With the former technique, we constructed a new discourse unit only of those tokens that formed a continuous span around the node-token that gave rise to the segment

Classifier	Macro-$F1$		Micro-$F1$		$F1_{tp,fp}$
	Mean	Std.Dev.	Mean	Std.Dev.	
`BitParSegmenter`	39.38%	±6.33%	97.49%	±0.21%	77%
`MateSegmenter`	47.8%	±6.13%	96.6%	±0.34%	80.05%

Table 5: Intrinsic evaluation of syntax-based classifiers.

(i.e., we neglected the part that was connected to this node via a non-projective edge). With the latter method, we collected all child tokens of the segment-generating node, and added the tokens that disrupted these children (i.e., those that occurred between the projective and the non-projective descendants) to the discourse segment as well. Examples of both approaches are shown in Figures 2a and 2b.

After testing both methods, we opted for the greedy technique, since it achieved slightly better results than the generous method, though the difference between the two approaches was not very big (the difference in P_k only amounted to 0.1 and the difference in π^*_{BED} only ran up to 0.2%).

4.3.3 Results

In order to train both classifiers and obtain segments based on their predictions, we applied 10-fold cross validation over the whole training corpus by successively splitting it into ten parts and subsequently training the classifier on nine of ten fractions, then applying the resulting system to the remaining part. We performed both an intrinsic and an extrinsic evaluation of the results.

In the intrinsic evaluation, we assessed how good each classifier was at predicting the correct segment classes (including NONE) for syntactic constituents and dependency nodes respectively. To do so, we estimated the mean and the standard deviation of the micro- and macro-averaged F-scores obtained in all 10 folds. The results of this evaluation are shown in Table 5.[9]

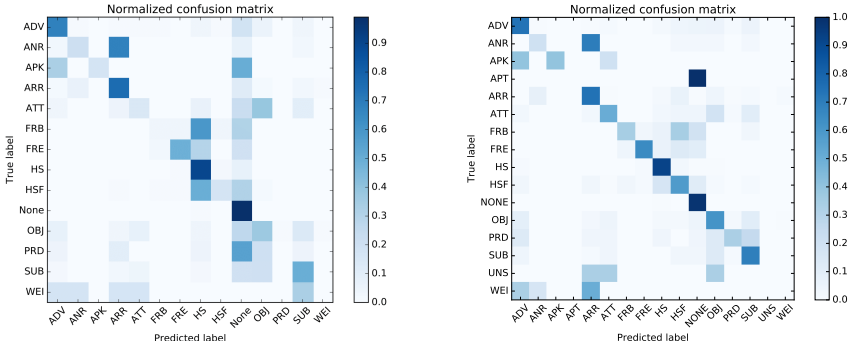
As one can see from the table, the dependency-based segmenter clearly outperforms the constituency-based one, even though our preliminary estimations of the respective upper bounds of these approaches suggested equal results. Furthermore, we also can observe a dramatic difference between the macro- and micro-averaged F-scores for both segmenters. This discrepancy can be explained by a skewed distribution of the segment classes in our corpus and different susceptibility of the two metrics to such

[9] All evaluations were carried out using cross validation with the released versions of the segmenters (v.0.0.1.dev1). To ease the alignment, we have also removed the backslash escapes of quotation marks and slashes that were introduced by BitPar. Note, however, that the pre-trained model delivered with the module was trained on the whole corpus and not on the nine cross-validation folds, so it might therefore produce slightly different results.

unbalanced data: while the micro-averaged F-score counts the total number of correct and wrong decisions, the macro-metric takes the average of the F-scores for predicting each particular segment class. The majority of the items in our data, however, have the gold class NONE which is also correctly predicted in most of the cases (as suggested by the micro-score). But since there are also many less frequent segment classes in the data set (some of which only occur a few times in our corpus), making even a few wrong predictions for these items leads to considerably lower macro-averaged results.

Another source of bias, which might significantly affect the evaluation, can be due to a skewed distribution of different gold classes across multiple folds. This problem was brought to our attention by one of the reviewers and particularized in the work of Forman and Scholz (2010). As a remedy for this issue, the authors suggest using an alternative average metric which they call $F1_{tp,fp}$ and which is calculated as a ratio $F1_{tp,fp} = \frac{2*TP}{2*TP+FP+FN}$, where TP, FP, and FN denote the total counts of true positive, false positive, and false negative predictions in all test folds of cross validation. As can be seen from the last column in Table 5, the results of this metric still correlate with other measurements and are situated between the micro- and macro-estimations.

To get a better intuition about the particular kinds of errors made by each segmenter, we also generated a confusion matrix of their wrong decisions (see Figure 3). Our goal was to see whether these two syntax-based approaches had complementary strengths and weaknesses or rather showed approximately the same behavior. As can be seen from the figure, the `BitParSegmenter` clearly tends to under-segment the input sentences. While this tendency is generally also observed for the `MateSegmenter`, it is much less acute there, and the confusion classes are spread more uniformly.



(a) `BitParSegmenter` **(b)** `MateSegmenter`

Figure 3: Confusion matrices for constituency- and dependency-based segmenters.

The classification results on their own are not very informative about the structural properties of the resulting segments, though. Indeed, the fact that some syntactic node will be correctly recognized as a segment-initiating item does not necessarily imply that

the recognized segment will be correctly integrated into the overall segment structure (if, for example, the levels of syntactic dependencies are confused in the syntax tree).

To check whether such discrepancies were present in our data, we performed an extrinsic evaluation of the resulting segmentation by applying the same agreement tests to the output of the automatic systems as we did for estimating the inter-annotator agreement between the human experts. In contrast to the previous IAA study, however, where we estimated the percentage of matching spans as a ratio between the spans annotated by both experts and the total number of spans annotated by just one annotator (whom we considered as the gold reference), this time, we had to introduce two metrics: matching $\text{spans}_{\text{pred}}$ and matching $\text{spans}_{\text{gold}}$. With the former benchmark, we computed the ratio of matching spans with respect to all *predicted* spans. With the latter metric, we estimated the percentage of matching segments with respect to the total number of *gold* segments as we did in the previous estimations.

The results of this evaluation are presented in Table 6. As can be seen from the table, the extrinsic figures still strongly correlate with the intrinsic macro-$F1$ scores. Furthermore, we can also observe that both automatic segmenters significantly outperform the baseline results and that the dependency-based approach generally yields better scores from both intrinsic and extrinsic perspectives.

Metric	Baseline		`BitParSegmenter`		`MateSegmenter`	
	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
P_k	30.99	±10.10	11.47	±4.39	5.37	±4.64
WinDiff	41.62	±15.23	18.61	±5.45	6.49	±5.42
π^*_{BED}	51.72	±11.49	64.43	±8.49	87.42	±9.77
α_{U}	47.54	±19.95	66.77	±13.66	89.12	±11.69
α^c_{U}	49.33	±17.56	59.98	±17.19	66.98	±19.53
parseval unlab. $F1$	25.12	±11.48	64.53	±17.39	78.52	±14.63
parseval lab. $F1$	25.12	±11.48	60.66	±17.53	72.07	±16.62
matching $\text{spans}_{\text{pred}}$	48.38		34.43		68.69	
matching $\text{spans}_{\text{gold}}$	50.87		48.16		78.96	
categories κ	65.24		72.07		74.91	

Table 6: Extrinsic evaluation of syntax-based segmenters.

4.4 Flat Segmentation

Instead of taking syntactic trees as input and trying to reconstruct discourse segments based on the predictions for their nodes, another viable alternative is to rely on plain sequences of tokens.

A clear advantage of this approach is that, in contrast to syntax-based systems, the underlying input data structure (token sequence), which serves as a basis for constructing the segments, is trivially guaranteed to be flawless and hence more amenable to a correct segmentation. At the same time, an incorrectly built syntactic tree (which not

infrequently happens in parsing) will almost inevitably lead to wrong segments even with correct classifiers.

A disadvantage of this method, however, is that plain token chains are much less suitable for hierarchical segmentation. Almost all approaches relying on token sequences therefore produce only a flat segmentation of the input sentences. Examples of such systems include the works proposed by Hernault et al. (2010a) and Bach et al. (2012). The system of Hernault et al. was, to the best of our knowledge, the first attempt to tackle the segmentation problem as a sequence labeling task based on Conditional Random Fields (Lafferty et al., 2001). This system operated on strings of tokens, but extensively utilized syntactic features obtained from parsers. Bach et al. (2012) later refined this method by first obtaining N-best sequences from a base CRF-classifier and then re-ranking those sequences judging by the properties of syntactic parse trees that bound or split potential segments.

The state-of-the art results for this type of processing were obtained by the method proposed by Feng and Hirst (2014). In their approach, the authors devised a two-pass system which first made initial guesses over a sequence of token pairs, predicting 'B' if there was a segment boundary in between two adjacent tokens and 'C' otherwise (see Figure 4). These guesses were subsequently corrected by another CRF-classifier (to be explained below).

For the purpose of our experiments, we adjusted the system of Feng and Hirst (2014) to the specifics of German text processing and tested it on our corpus in the same cross-validation fashion as we did for the hierarchical systems explained above. For this purpose, we first converted hierarchical discourse structures annotated in our data to a flat segmentation, as explained earlier and as illustrated in Figure 4. We then applied the method of Feng and Hirst (2014) and separately tested its one- and two-pass variants.

Input sentence: `(HS Doch das Angebot , (ATT nahe der Haustür den Personennahverkehr nutzen zu können , ) wollen nicht alle . )`

Label sequence: `C C C B C C C C C C C C B C C C`

Figure 4: Flattening of a hierarchical segment structure.

4.4.1 One-pass Model

The one-pass variant corresponds to a plain CRF classifier that takes a sequence of feature representations for token pairs and returns the most probable assignment of segment boundaries for this sequence.

Following the original approach, we used the same set of feature attributes for each pair of adjacent tokens:

- the part-of-speech tags and the lemmas of both tokens;
- Boolean features indicating whether the first or the last token of the pair were located at the beginning or the end of a sentence;
- The part-of-speech tags of the top dependency nodes for which the first and the last token of the pair were the left- and the right-most children respectively;[10]
- the height of the sub-trees whose left- and right-most children were the first and the last token of the pair respectively;
- the top production rule of the largest syntactic constituents starting from the first or ending with the last token of the pair;
- the same set of features for the left and right neighbor token pairs.

4.4.2 Two-pass Model

After the first stage is complete, a second pass of the algorithm corrects the hypothesized segment boundaries by effectively applying the same CRF-method and the same set of token-pair features as in the first pass, plus taking into account the global properties of the potential segments such as:

- the part-of-speech tags and the lemmas of the tokens located at the left / right boundary of the enclosing discourse segment;
- the distance to the nearest left / right segment boundary;
- the number of syntactic nodes between the token pair in question and its nearest discourse segment boundaries;
- the part-of-speech tag of the top node of the lowest sub-tree that encompasses all tokens between the respective token pair and its left / right neighboring segment boundary.

4.4.3 Results

To evaluate the one- and two-pass variants of this approach, we applied the same 10-fold cross validation strategy as we did for the syntax-based hierarchical methods. The results of the macro- and micro-averaged F-scores for this two-class classification task are shown in Table 7. We also performed an extrinsic evaluation of the resulting segment structures whose results are presented in Table 8.

[10] In this regard, our features slightly differ from the ones adopted by Feng and Hirst (2014). Since syntactic features used in their work were obtained from the output of the Stanford Parser (Klein and Manning, 2003), the authors used syntactic labels of tree constituents at this point. We, however, operate on the output of the Mate parser and use the part-of-speech tags of the top nodes instead, since this parser does not provide constituents.

Classifier	Macro-$F1$		Micro-$F1$		$F1_{tp,fp}$
	Mean	Std.Dev.	Mean	Std.Dev.	
One-pass model	87%	±1%	97.67%	±0.13%	76.33
Two-pass model	93.19%	±1.5%	98.66%	±0.26%	88.26

Table 7: Intrinsic evaluation of CRF-based classifiers.

Metric	Baseline		One-pass Model		Two-pass Model	
	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
P_k	30.99	±10.10	9.24	±5.80	4.29	±4.03
WinDiff	41.62	±15.23	10.91	±6.69	5.14	±4.61
π^*_{BED}	51.72	±11.49	81.03	±10.08	89.96	±8.73
α_{U}	47.54	±19.95	81.69	±14.32	91.18	±10.03
α^c_{U}	49.33	±17.56	46.67	±18.89	46.39	±19.88
parseval unlab. $F1$	25.12	±11.48	74.25	±14.37	77.62	±15.17
parseval lab. $F1$	25.12	±11.48	70.46	±16.47	73.23	±16.86
matching $\text{spans}_{\text{pred}}$	48.38		53.47		59.99	
matching $\text{spans}_{\text{gold}}$	50.87		74.08		84.92	
categories κ	65.24		23.67		18.57	

Table 8: Extrinsic evaluation of CRF-based segmenters.

As can be seen from the table, even the one-pass model outperforms the baseline method by a factor of three in terms of the P_k measure. This improvement is even bigger when measured with WinDiff, where the error drop is quadruple. Estimations with other metrics yield consistent results: π^*_{BED} is improved by 29.31%, whereas α_{U} is increased by 34.15%. The most drastic quality boost, however, can be observed for the parseval measurements: here the unlabeled F-score changes from 25.12% to almost 75%, and the labeled metric surpasses the considerably high 70% landmark.

These results are further improved by the two-pass classifier, which not only outperforms the one-pass approach but also yields better scores for P_k, WinDiff, π^*_{BED}, α_{U}, and the labeled parseval $F1$ than the tree-based MateSegmenter. Two-pass CRFs are still approximately on par with the mate system in terms of the unlabeled parseval $F1$, but perform significantly worse than that when measured with α^c_{U}. An obvious explanation for this is that the latter metric puts much weight on the correct prediction of segment categories – a part which we deliberately sacrificed when flattening the segment structures. Nevertheless, we consider it as an interesting finding that a plain sequence-based segmentation method forms a viable alternative to the tree-based approaches in many other regards.

5 Summary

The central contribution of this paper is an implemented, comparative approach to discourse segmentation of German texts. We provide a thorough discussion of the evaluation problem for flat and hierarchical segmentation, and measure our inter-annotator agreement and the performance of the automatic approaches by various means. For applications where a hierarchical and possibly also labeled segmentation is advantageous, we offer classifiers operating on the output of state-of-the-art German syntax parsers (mate and BitPar). Our results show an advantage for mate as the basis of a segmenter, but this could be due to the absence of POS tags in the BitPar output, which thus did not enter our feature set. We leave it to future work to determine whether a POS-enhanced version of the feature set would improve the results (or, conversely, whether the mate results would deteriorate if the POS features were left out). To our knowledge, this is the first comparison of the two linguistic parsing strategies regarding their suitability for a subsequent discourse segmentation step, and our error analysis indicates that the difference in the results stems from a tendency to under-segmentation on the side of BitPar. The general technique we use resembles that of Soricut and Marcu (2003). We do not comment on the relationship between the evaluation results, because syntactic parsing of English and German are clearly not of the same difficulty, so there is little point in comparing our numbers to those of SPADE.

When hierarchy and labels are not needed, a CRF model yields very good results. Our implementation followed that of Feng and Hirst (2014), but we made a number of adaptations that were necessary for applying this technique to German. Again, we do not compare the German versus English results, but we notice that for the German data, the CRF approach yields better performance than the tree classification techniques; but that is probably not surprising, because the task of flat segmentation is somewhat easier.

Our three implementations are freely available online[11] and thus constitute – to the best of our knowledge – the first re-usable modules for this discourse processing task for German. Likewise, our corpus of annotated hierarchical and labeled discourse segments as well as the accompanying guidelines for this corpus are also released[12] and can be freely used for research purposes.

References

Afantenos, S. D., Denis, P., Muller, P., and Danlos, L. (2010). Learning recursive segments for discourse parsing. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., Rosner, M., and Tapias, D., editors, *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010, 17-23 May 2010, Valletta, Malta*. European Language Resources Association.

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

[11] https://github.com/discourse-lab/DiscourseSegmenter
[12] http://angcl.ling.uni-potsdam.de/resources/pcc.html

Bach, N. X., Nguyen, M. L., and Shimazu, A. (2012). A reranking model for discourse segmentation using subtree features. In *Proceedings of the SIGDIAL 2012 Conference, The 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 5-6 July 2012, Seoul National University, Seoul, South Korea*, pages 160–168. The Association for Computer Linguistics.

Beeferman, D., Berger, A., and Lafferty, J. (1999). Statistical models for text segmentation. *Mach. Learn.*, 34(1-3):177–210.

Black, E., Abney, S. P., Flickenger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J. L., Liberman, M., Marcus, M. P., Roukos, S., Santorini, B., and Strzalkowski, T. (1991). A procedure for quantitatively comparing the syntactic coverage of english grammars. In *Speech and Natural Language, Proceedings of a Workshop held at Pacific Grove, California, USA, February 19-22. 1991*. Morgan Kaufmann.

Bohnet, B. (2010). Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 89–97, Stroudsburg, PA, USA. Association for Computational Linguistics.

Brants, S., Dipper, S., Eisenberg, P., Hansen, S., König, E., Lezius, W., Rohrer, C., Smith, G., and Uszkoreit, H. (2004). Tiger: Linguistic interpretation of a german corpus. *Research on Language and Computation*, 2(4):597–620.

Breiman, L. et al. (1984). *Classification and Regression Trees*. Chapman & Hall, New York.

Bußmann, H. (2002). *Lexikon der Sprachwissenschaft*. Kröner, Stuttgart.

Carlson, L. and Marcu, D. (2001). *Discourse tagging manual. Technical report*. Univ. of Southern California/ISI.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Crammer, K. and Singer, Y. (2002). On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Feng, V. W. and Hirst, G. (2014). Two-pass discourse segmentation with pairing and global features. *CoRR*, abs/1407.8215.

Fisher, S. and Roark, B. (2007). The utility of parse-derived features for automatic discourse segmentation. In *Proc. of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 488–495, Prague, Czech Republic. Association for Computational Linguistics.

Fix, E. and Hodges, J. L. J. (1989). Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review*, 57(3):238–247.

Forman, G. and Scholz, M. (2010). Apples-to-apples in cross-validation studies: pitfalls in classifier performance measurement. *SIGKDD Explorations*, 12(1):49–57.

Fournier, C. (2013). Evaluating text segmentation using boundary edit distance. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 1702–1712. The Association for Computer Linguistics.

Grosz, B., Joshi, A., and Weinstein, S. (1995). Centering: A framework for modelling the local coherence of discourse. *Computational Linguistics*, 21(2):203–226.

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

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

Hernault, H., Bollegala, D., and Ishizuka, M. (2010a). A sequential model for discourse segmentation. In Gelbukh, A. F., editor, *Computational Linguistics and Intelligent Text Processing, 11th International Conference, CICLing 2010, Iasi, Romania, March 21-27, 2010. Proceedings*, volume 6008 of *Lecture Notes in Computer Science*, pages 315–326. Springer.

Hernault, H., Prendinger, H., duVerle, D., and Ishizuka, M. (2010b). Hilda: A discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1–33.

Ji, Y. and Eisenstein, J. (2014). Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, Baltimore/MD*, pages 13–24.

Klein, D. and Manning, C. D. (2003). Accurate unlexicalized parsing. In Hinrichs, E. W. and Roth, D., editors, *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 7-12 July 2003, Sapporo Convention Center, Sapporo, Japan.*, pages 423–430. ACL.

Krippendorff, K. (1995). On the reliability of unitizing continuous data. *Sociological Methodology*, 25:47–76.

Krippendorff, K. (2004). Measuring the reliability of qualitative text analysis data. *Quality & Quantity*, 38:787–800.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Brodley, C. E. and Danyluk, A. P., editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Lappin, S. and Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics*, 20(4):535–561.

Le Thanh, H., Abeysinghe, G., and Huyck, C. (2004). Generating discourse structures for written text. In *Proc. of the 20th International Conference on Computational Linguistics (COLING)*, pages 329–335, Geneva/Switzerland.

Liaw, A. and Wiener, M. (2002). Classification and regression by randomforest. *R News*, 2(3):18–22.

Lüngen, H., Puskás, C., Bärenfänger, M., Hilbert, M., and Lobin, H. (2006). Discourse segmentation of german written texts. In Salakoski, T., Ginter, F., Pyysalo, S., and Pahikkala, T., editors, *Advances in Natural Language Processing, 5th International Conference on NLP, FinTAL 2006, Turku, Finland, August 23-25, 2006, Proceedings*, volume 4139 of *Lecture Notes in Computer Science*, pages 245–256. Springer.

Mann, W. and Thompson, S. (1988). Rhetorical structure theory: Towards a functional theory of text organization. *TEXT*, 8:243–281.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization*. MIT Press, Cambridge/MA.

Meyer, C. M., Mieskes, M., Stab, C., and Gurevych, I. (2014). Dkpro agreement: An open-source java library for measuring inter-rater agreement. In Tounsi, L. and Rak, R., editors, *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations (COLING)*, pages 105–109, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Miltsakaki, E. (2002). Toward an aposynthesis of topic continuity and intrasentential anaphora. *Computational Linguistics*, 28(3):319–355.

Mosegaard-Hanse, M.-B. (1998). *The Function of Discourse Particles: A Study with Special Reference to Spoken Standard French*. John Benjamins, Amsterdam and Philadelphia.

Muller, P., Afantenos, S., Denis, P., and Asher, N. (2012). Constrained decoding for text-level discourse parsing. In *Proc. of the International Conference on Computational Linguistics (COLING)*, Mumbai, India.

Needleman, S. and Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Pevzner, L. and Hearst, M. A. (2002). A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36.

Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The penn discourse treebank 2.0. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association.

Schapire, R. E. and Singer, Y. (2000). Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

Schmid, H. (2004). Efficient parsing of highly ambiguous context-free grammars with bit vectors. In *COLING 2004, 20th International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2004, Geneva, Switzerland*.

Schmidt, T., Wörner, K., Hedeland, H., and Lehmberg, T. (2011). New and future developments in EXMARaLDA. In Schmidt, T. and Wörner, K., editors, *Multilingual Resources and Multilingual Applications. Proceedings of GSCL Conference 2011 Hamburg*.

Schmitt, H. (2000). *Zur Illokutionsanalyse monologischer Texte*. Peter Lang, Frankfurt.

Soricut, R. and Marcu, D. (2003). Sentence-level discourse parsing using syntactic and lexical information. In *Proc. of the Human Language Technology Conference of the North American Chapter of the ACL*, pages 149–156, Edmonton/Canada.

Sporleder, C. and Lapata, M. (2005). Discourse chunking and its application to sentence compression. In *Proc. of the HLT/EMNLP Conference*, pages 257–264, Vancouver.

Stede, M. (2011). *Discourse Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Stede, M., Mamprin, S., and Peldszus, A. (2015). Diskurssegmentierung. In Stede, M., editor, *Handbuch Textannotation: Potsdamer Kommentarkorpus 2.0*, volume 8 of *Potsdam Cognitive Science Series*, pages 23–44. Universitätsverlag Potsdam.

Stede, M. and Neumann, A. (2014). Potsdam Commentary Corpus 2.0: Annotation for discourse research. In *Proc. of the International Conference on Language Resources and Evaluation (LREC)*, Reykjavik.

Stede, M. and Peldszus, A. (2012). The role of illocutionary status in the usage conditions of causal connectives and in coherence relations. *Journal of Pragmatics*, 44(2):214–229.

Taboada, M. and Zabala, L. H. (2008). Deciding on units of analysis within centering theory. *Corpus Linguistics and Linguistic Theory*, 4(1):63–108.

Tetreault, J. (2001). A corpus-based evaluation of centering and pronoun resolution. *Computational Linguistics*, 27(4):507–520.

Tofiloski, M., Brooke, J., and Taboada, M. (2009). A syntactic and lexical-based discourse segmenter. In *Proc. of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP (Short Papers)*, pages 77–80, Suntec, Singapore. Association for Computational Linguistics.

Versley, Y. and Gastel, A. (2012). Linguistic tests for discourse relations in the TüBa-D/Z corpus of written German. *Dialogue & Discourse*, 4 (2).

Webber, B. L., Egg, M., and Kordoni, V. (2012). Discourse structure and language technology. *Natural Language Engineering*, 18(4):437–490.