Learning Recursive Segments for Discourse Parsing

Stergos Afantenos*, Pascal Denis†, Philippe Muller* †, Laurence Danlos†

*Institut de recherche en informatique de Toulouse (IRIT)
Université Paul Sabatier
118 route de Narbonne, 31062 Toulouse Cedex 9, France
{stergos.afantenos, philippe.muller}@irit.fr

†Equipe-Projet Alpage
INRIA & Université Paris 7
30 rue Château des Rentiers, 75013 Paris, France
laurence.danlos@linguist.jussieu.fr, pascal.denis@inria.fr

Abstract

Automatically detecting discourse segments is an important preliminary step towards full discourse parsing. Previous research on discourse segmentation have relied on the assumption that elementary discourse units (EDUs) in a document always form a linear sequence (i.e., they can never be nested). Unfortunately, this assumption turns out to be too strong, for some theories of discourse like SDRT allows for nested discourse units. In this paper, we present a simple approach to discourse segmentation that is able to produce nested EDUs. Our approach builds on standard multi-class classification techniques combined with a simple repairing heuristic that enforces global coherence. Our system was developed and evaluated on the first round of annotations provided by the French Annodis project (an ongoing effort to create a discourse bank for French). Cross-validated on only 47 documents ($1,445$ EDUs), our system achieves encouraging performance results with an F-score of 73% for finding EDUs.

1. Introduction

Discourse parsing is the analysis of a text from a global, structural perspective: how parts of a discourse contribute to its global interpretation, accounting for semantic and pragmatic effects beyond simple sentence concatenation. This task consists in two main steps: (i) finding the elementary discourse units (henceforth EDUs), and (ii) organizing them in a way that make explicit their functional (aka rhetorical) relations. Popular theories of discourse include Rhetorical Structure Theory (RST) (Mann and Thompson, 1987), Discourse Lexicalized Tree-Adjoining Grammar (DLTAG) (Webber, 2004), Segmented Discourse Representation Theory (SDRT) (Asher, 1993). Each of these theoretical frameworks has been at the center of important corpus building efforts, see (Carlson et al., 2003; Prasad et al., 2004; Baldridge et al., 2007) respectively. In the present work, we focus on the first step, namely segmenting a discourse into EDUs, within a larger project aiming at building an SDRT discourse corpus of French texts.

In addition to being a necessary step in discourse parsing, discourse segmentation, could also be useful as a stand-alone application for a variety of other tasks where EDUs could provide simpler input than sentences. Examples of such tasks are: automatic summarization and sentence compression, bitext alignment, translation, chunking/syntactic parsing.

The first discourse segmentation system dates back to the rule-based work of (Ejerhed, 1996), which was a component in the RST-based parser of (Marcu, 2000). More recently, (Tofiloski et al., 2009) tested a rule-based segmenter on top of a syntactic parser, achieving F-score of 80-85% in segment boundary identification on a slightly modified RST corpus. Machine learning based segmentation systems have also been proposed, notably by (Soricut and Marcu, 2003), (Sporleder and Lapata, 2005) and (Fisher and Roark, 2007). The latter report F-score of 90.5% in boundary detection (and 85.3% in correct bracketing) on the RST corpus.

Discourse segmentation is to large extent theory dependent, for different theories make different assumptions on what EDUs can be. Carried out on the RST corpus, previous work on discourse segmentation has exploited an important particularity of this corpus: namely, the fact that it does not have any embedded EDUs. These approaches have been able to recast discourse segmentation as a *binary* classification problem: that is, each text position (token or token separator) is either a segment boundary or not. By contrast to RST, other theories like SDRT allows for embedded EDUs: embedding is used to encode modifying clauses like non restrictive relatives (including reduced relatives) and appositions. As will be discussed in Section 2., our SDRT-based corpus does contain close to 10% of nested EDUs.

Predicting nested structures introduces additional difficulties, in particular that of outputting a coherent, balanced bracketing. This characteristic renders discourse segmentation akin to syntactic clause boundary identification (CBI), a task which has received some attention from the CL community. The main approach to CBI is to classify tokens into *three* classes for clause start, end, or inside. The best results obtained during the CoNLL-2001 campaign were 89-90% for boundary detection and 81.73% for correct clause identification (correct guessing of start and end), with boosted decision trees (Carreras and Màrquez, 2001).

We have adapted this general setting to the problem of discourse segmentation, with possible embedded segments, and applied it to a corpus of French discourses, part of an on-going corpus building project.

2. Data and Evaluation

2.1. Corpus

The corpus we use has been developed as part of the Annodis project[1], an on-going effort to annotate French discourses from various genres with both top-level, typographic structures and local coherence relations. About 100-150 texts are being segmented and annotated with coherence relations. These documents are drawn mainly for wikipedia articles and from *L'est républicain* newspaper [2]. Text length varies from 300 to 900 tokens. Annotations are performed by pairs of human annotators in a two-step process: (i) individual annotations, and (ii) adjudication. The present work considers the 47 texts that have undergone validation. The average number of EDUs per document in this set is 33.

Segments typically correspond to verbal clauses, but also other syntactic units describing eventualities (such as prepositional phrases), adjuncts such as appositions or cleft constructions with discursive long-range effects such as frame adverbials. A particularity of the discourse units in Annodis is that they can be embedded in one another, as in example in figure 1 (brackets mark segmentation).

In this example, the EDUs π_1 *mondialement connues,* and π_2 *donc difficilement écoulables,* are nested within the the main, discontinuous EDU π_0 *Ces pièces avaient été repérées chez un riche amateur nippon.*

2.2. Evaluation

Discourse segmentation evaluation is typically performed in terms of precision, recall, and F-score for segment boundaries (Soricut and Marcu, 2003; Fisher and Roark, 2007; Sporleder and Lapata, 2005). Previous work differ as to whether they include sentence boundaries (e.g., (Soricut and Marcu, 2003) are only interested in sentence-internal segmentation) and whether they additionally require labeling of the segments (Sporleder and Lapata, 2005).

Since the type of segmentation we produce includes nested EDUs, we have to resort to another type of evaluation. For this paper, we use the three metrics commonly used for evaluating clause detection: (i) precision, recall, and F-score for segment start position, (ii) precision, recall, and F-score for segment end position, and (iii) precision, recall, and F-score for complete segments. These metrics correspond to three tasks included in the CoNLL 2001 shared task.

[1] `http://w3.erss.univ-tlse2.fr/textes/pagespersos/annodis/`

[2] `http://www.cnrtl.fr/corpus/estrepublicain/`

[Ces pièces, [mondialement connues,]$_{\pi_1}$ [donc difficilement écoulables,]$_{\pi_2}$ avaient été repérées chez un riche amateur nippon]$_{\pi_0}$

[The pieces, [worldwide famous,]$_{\pi_1}$ [thus hard to resell,]$_{\pi_2}$ had been located at a rich japanese art lover's]$_{\pi_0}$

Figure 1: A discourse segmentation from the Annodis corpus.

3. Approach

3.1. Classification Model

Like previous approaches to discourse segmentation and CBI, we cast the task of EDU identification as a classification problem. Specifically, we built a *four*-class classifier that maps each token w_i in a discourse $w_1, \ldots, w_n$ to one of the following boundary types $B = \{\texttt{left}, \texttt{right}, \texttt{both}, \texttt{nothing}\}$. These correspond to the different bracketing configurations found in our corpus, respectively (i) w_i opens a segment, (ii) w_i ends a segment, (iii) w_i is a single-token segment, and (iv) none of the above. If we take the beginning of the example in 2.1., *[Ces pièces, [mondialement connues,] Ces* and *mondialement* would be classified as `left`, the last comma as `right`, and all other tokens as `nothing`.

For our classifier, we used a regularized maximum entropy (MaxEnt, for short) model (Berger et al., 1996). In MaxEnt, the parameters of an exponential model of the following form are estimated:

$$P(b|t) = \frac{1}{Z(b)} \exp\left(\sum_{i=1}^{m} w_i f_i(t, b)\right)$$

where t represents the current token and b the outcome (i.e., the type of boundary). Each token t is encoded as a vector of m indicator features f_i. There is one weight/parameter w_i for each feature f_i that predicts its classification behavior. Finally, $Z(b)$ is a normalization factor over the different class labels (in this case, the 4 boundary types), which guarantees that the model outputs probabilities.

In MaxEnt, the values for the different parameters $\hat{w}$ are obtained by maximizing the log-likelihood of the training data T with respect to the model (Berger et al., 1996):

$$\hat{w} = \underset{w}{\mathrm{argmax}} \sum_{i}^{T} \log P(b^{(i)}|t^{(i)})$$

Various algorithms have been proposed for performing parameter estimation (see (Malouf, 2002) for a comparison). Here, we used the Limited Memory Variable Metric Algorithm implemented in the MegaM package.[3] We used the default regularization prior that is used in MegaM.

3.2. Feature Set

Our feature set relies on two main sources of information. The first source is a list of *lexical markers*, containing discourse connectives and a few indirect speech report verbs that are likely to introduce discourse units. Specifically, we created boolean features that check whether the token is part of connectives (resp. verbs) in our list of markers.

The other information source is (morpho-)syntactic, drawn from the automatic analysis provided by the Macaon chunker (Nasr and Volanschi, 2006) and the SYNTEX dependency parser (Bourigault et al., 2005). Using these two analyzers, we extract for each token: its lemma, its part-of-speech (POS) tag, its chunk tag, its dependency path to the root element (as well as "sub-paths" of length 1-3), and its inbound dependencies. In addition, we also capture the linear position of the word in a sentence (we used quantized values ranging from 1-100). These feature templates were also applied to the surrounding words in a window of 3 words to the left and right.

Two more feature families were added. The first concerns the outward chunk sequence for each token; that is, given that a token is embedded in a sequence of chunks, we start from the innermost chunk tag and we go out all the way to the outermost. These features exploit the fact that Macaon provides some level of embedding in its chunks. The second feature family concerns all

[3] Available from `http://www.cs.utah.edu/~hal/megam/`.

the n-gramms $1 < n \leq 6$ for which the token is included and their span does not exceed the boundaries of the current sentence. A synoptic table with the entire feature set we used is shown in table 1.

3.3. Resampling

The distribution of boundary types is heavily skewed towards nothing (about 12.000 instances against about 1400 for each left and right), which suggests that resampling our data toward a more uniform distribution might lead to better classification accuracy, and in turn to better EDU segmentation.

The resampling method we used directly exploits the syntactic chunk boundaries as found by the Macaon chunker. It is based on the observation that EDU boundaries in a large majority of cases coincide with chunk boundaries. The output of Macaon was used in the following ways. First, we decided to replace the decisions on sentence boundary tokens with the decisions that Macaon provides. In other words, sentence boundary tokens, as given by Macaon, were ignored during training; they were tagged as left and right respectively during test. Second, we also removed from training tokens that were *strictly* inside chunks (that is, tokens that are inside a chunk but doesn't correspond to its beginning or end). At test, these tokens were assigned the nothing class. All remaining tokens were used for training and follow the classification decoding at test. After those modifications, the class distribution was around 9200 instances for the class nothing, while the rest of the classes had around 1400 instances.

3.4. Enforcing coherence

Casting segmentation as a series of local classifications has two major drawbacks. First, the segmentation decision at a token is highly dependent from the decisions on neighboring tokens. Secondly, unrelated local decisions do not guarantee the well-formedness of the segmentation of a sentence, since we allow for embedded segments. For instance, the number of beginning of embedded segments must obviously match the number of endings.

A straightforward way to capture Markovian dependencies between segmentation labels is to encode previous labels as features of the model, in combination with a Viterbi decoding. Unfortunately, we found during development that this strategy degrades segmentation performance, probably due to the sparsity of the boundary labels.[4]

To tackle the problem of ensuring a coherent bracketing, we propose a specific post-processing on the outputs of the classifier. In particular, we apply heuristic repair techniques (adding/deleting boundaries) to yield a well-formed sentence segmentation. A simple technique proved efficient enough: we scanned sentences token by token from beginning to end, while keeping track of the depth of the current EDU embedding. If the depth is 0 before the end of a sentence, it means we found a stranded token, that is then reclassified as left; this rebalances the number of left and right. Dually, we reversed the sequences to reclassify remaining out-of-segment tokens as right. This heuristic is illustrated in figure 2. In the future we plan to apply local optimization techniques under well-formedness constraints, to repair segmentations while better preserving the probability on each decision.

4. Experiments and Results

We present two sets of scores, one without post-processing and one with post-processing. We did a 10-fold cross-validation on the sentences contained in the 47 documents of the corpus. We used the three metrics for segmentation evaluation discussed in section 2.; we also report precision, recall, and F-score for the both boundary class.

Table 2 (resp. table 3) reports the performance scores of the "classifier-only" system (resp. "classifier+post-processing" system) for the first series of experiments. In terms of overall classification performance, both systems perform similarly, but the second system improves on the three boundary classes {left, right, both}. The main source of improvement comes from recall, which suggests that our heuristics recover boundaries that were missed by the classifier.

Before post-processing, the proportion of not well-formed segmentations on the (recognized)

[4] Similar findings are reported by (Fisher and Roark, 2007).

Feature	Description
Lemma	the token's lemma (Syntex)
POS	Part of speech (Macaon)
Grammatical category	the main grammatical category of the token: V, N, P, etc. (Syntex)
start of a discourse marker	boolean, indicating whether the tokens starts a discourse marker
indirect speech report verb	boolean, indicating whether the token belongs to a predefined list of verbs.
dependency path	the dependency path from the word towards the root, limited to distance 3 (Syntex)
inbound dependencies	the inbound dependency relations for each token (Syntex)
syntactic projections	the number of times that the token is at the start, end or middle of an NP, VP, PP projection (Syntex)
distance from sentence boundaries	the relative distance from each of the sentence boundaries
context 3-grams	the lemma and POS 3-grams before and after the token (Syntex & Macaon)
chunk start/end	boolean features; token coincides with a chunk start/end (Macaon)
outward chunk tag sequence	the sequence of chunk tags from the innermost to the outermost chunk (Macaon)
context n-gramms	all the n-gramms ($1 < n \leq 6$) that include the token and do not exceed the limits of the sentence. The n-grams include Lemmas (Syntex), POS tags (Macaon) and Chunk tags (Macaon)

Table 1: Features used for the second approach (including chunks).

Input from classifier:
[The pieces,] worldwide famous,] thus hard to resell,] had been located [at a rich japanese art lover's]

First pass left-to-right:
[The pieces,] [worldwide famous,] [thus hard to resell,] [had been located [at a rich japanese art lover's]

First pass right-to-left: *[The pieces,] [worldwide famous,] [thus hard to resell,] [had been located] [at a rich japanese art lover's]*

Figure 2: Example repairing of a not well-formed segmentation with additions underlined. The sentence can now be compared to the reference, cf figure 1.

sentences is 35%, our post-processing heuristics yield 98% well-formed segmentations. The impact on precision/recall is shown in table 3.

The overall bad performance on `both` is due to the lack of data for this class: there are less than 20 examples in the entire corpus. When it comes to the segment evaluation, again the best results were achieved by the second approach which managed to correctly identify 73% of the manually annotated segments. These results are slightly less, but close to, the best results obtained by systems on the CBI task. Of course, the main reason post-processing boosts the EDU score is that a third more of the sentences are now evaluated, since they are well-formed. But the decline in precision is much less than the gain in recall.

4.1. Learning Curve

For their RST EDU segmentation experiments, Fisher and Roark (2007) have been using the RST-DT corpus which consists of a total of 385 documents (176,000 tokens). Carreras and

Class	Recall	Precision	F-measure
Left	0.845	0.891	0.868
Right	0.881	0.925	0.902
Both	0.684	0.812	0.742
EDUs	0.427	0.880	0.575

Table 2: Evaluation without post-processing.

Class	Recall	Precision	F-measure
Left	0.876	0.880	0.878
Right	0.885	0.889	0.888
Both	0.684	1.0	0.812
EDUs	0.719	0.748	0.733

Table 3: Evaluation with post-processing.

Màrquez (2001) have used the CoNLL 2001 corpus for the task of clause boundaries identification: this corpus includes sections $15 - 18$ of the Penn Treebank for training $(211,727$ tokens) and section 20 for test $(47,377$ tokens). In contrast to those approaches we have worked, as mentioned in section 2. we have been working with 47 validated documents (14384 tokens) from the Annodis project. Given that the number of documents that we have been working with is limited, at least in comparison with other approaches, we have calculated the learning curve for this number of documents in order to understand how the learning procedure will be influenced once we have the totality of our documents annotated. As mentioned in section 2. the total number of documents expected will be in the range of 100 to 150.

In order to calculate our learning curve, we divided our corpus into 9 different learning sets, starting from 5 random documents and incrementally adding 5 random documents into each learning set. For each such set we performed a ten-fold cross-validation, in the same way as described in section 4., using the feature set shown in table 1.

The learning curve is shown in figure 3. As it can be seen from this figure, the curves for both classes (`left` and `right`) grow regularly between sets 5 to 30 while it seems to plateau between sets 30 and 40 only to start going up again during the last set of documents. In general, it seems that the addition of more documents will only slightly increase the performance of our approach.

5. Conclusions and Future Work

Discourse segmentation is a crucial preprocessing stage for discourse analysis, and the global reliability of discourse parsing is heavily determined by success at this level. We have proposed a simple approach combining a 3-class classifier with a post-processing heuristics that achieve reasonable results, although the data available at the moment is limited. We need to see how this generalizes to the whole corpus, and to check how dependent it is on the nature of the corpus (newspaper articles and encyclopedia article). Another angle we plan to investigate is the usefulness of a non-perfect segmentation to help annotators start discourse annotation. Given the cost of human annotation of discourse, saving time on the segmentation would be a boost to annotators productivity, provided we verify that time spent is roughly proportional to the number of errors in the automated preprocessing; that hypothesis is not necessarily true, and there might be a threshold on the precision of the processing that is acceptable. Mainly, the ideal trade-off between precision and recall remains to be investigated.

6. References

N. Asher. 1993. *Reference to Abstract Objects in Discourse*. Kluwer.

J. Baldridge, N. Asher, and J. Hunter. 2007. Annotation for and robust parsing of discourse structure on unrestricted texts. *Zeitschrift fur Sprachwissenschaft*, 26:213–239.

A. Berger, S. Della Pietra, and V. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

D. Bourigault, C. Fabre, C. Frérot, M.-P. Jacques, and S. Ozdowska. 2005. Syntex, analyseur syntaxique de corpus. In *Actes des 12èmes journées sur le Traitement Automatique des Langues Naturelles*.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In Jan van Kuppevelt and Ronnie Smith, editors, *Current Directions in Discourse and Dialogue*, pages 85–112. Kluwer Academic Publishers.

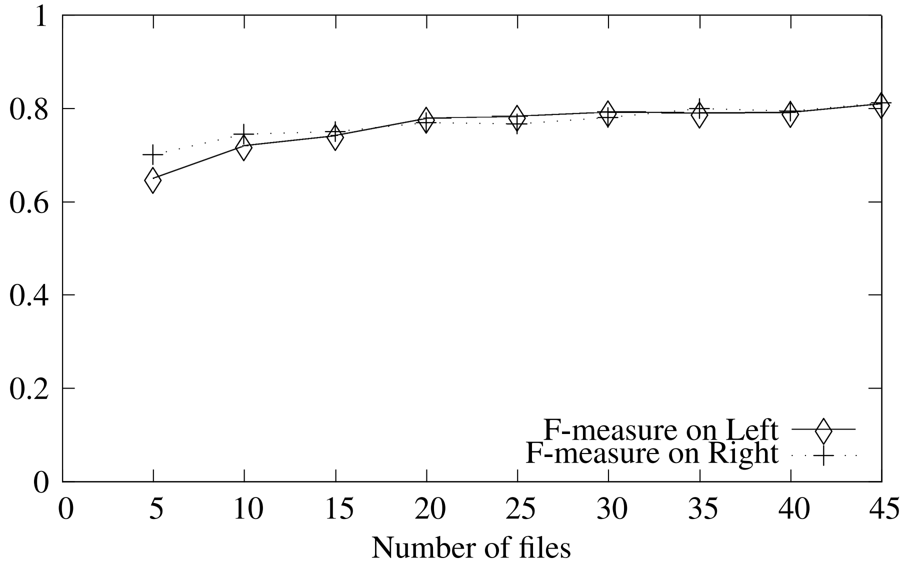

Figure 3: Learning curve for the series of experiments including chunking.

Xavier Carreras and Luís Màrquez. 2001. Boosting trees for clause splitting. In Walter Daelemans and Rémi Zajac, editors, *Proceedings of CoNLL-2001*, pages 73–75. Toulouse, France.

Eva Ejerhed. 1996. Finite state segmentation of discourse into clauses. In *Proceedings of the ECAI '96 Workshop on Extended finite state models of language*, Budapest.

Seeger Fisher and Brian Roark. 2007. The utility of parse-derived features for automatic discourse segmentation. In *ACL*.

Robert Malouf. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Workshop on Natural Language Learning*, pages 49–55, Taipei, Taiwan.

W. Mann and S. Thompson. 1987. Rhetorical structure theory : a theory of text organization. Technical report, Information Science Institute.

Daniel Marcu. 2000. The rhetorical parsing of unrestricted texts: a surface-based approach. 26(3):395–448.

Alexis Nasr and Alexandra Volanschi, 2006. *Finite-State Methods and Natural Language Processing*, volume 4002 of *Lecture Notes in Computer Science*, chapter Integrating a Part of Speech Tagger and a Chunker Implemented as Weighted Finite-State Machines, pages 167–178. Springer.

Rashmi Prasad, Eleni Miltsakaki, Aravind Joshi, and Bonnie Webber. 2004. Annotation and data mining of the penn discourse treebank. In *Proceedings of the ACL Workshop on Discourse Annotation*, Barcelona, Spain.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *HLT/NAACL*, Edmonton, Canada.

Caroline Sporleder and Mirella Lapata. 2005. Discourse chunking and its application to sentence compression. In *HLT/EMNLP*.

Milan Tofiloski, Julian Brooke, and Maite Taboada. 2009. A syntactic and lexical-based discourse segmenter. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 77–80, Suntec, Singapore, August. Association for Computational Linguistics.

Bonnie L. Webber. 2004. D-ltag: extending lexicalized tag to discourse. *Cognitive Science*, 28(5):751–779.