A Syntactic and Lexical-Based Discourse Segmenter

Milan Tofiloski
School of Computing Science
Simon Fraser University
Burnaby, BC, Canada
mta45@sfu.ca

Julian Brooke
Department of Linguistics
Simon Fraser University
Burnaby, BC, Canada
jab18@sfu.ca

Maite Taboada
Department of Linguistics
Simon Fraser University
Burnaby, BC, Canada
mtaboada@sfu.ca

Abstract

We present a syntactic and lexically based discourse segmenter (SLSeg) that is designed to avoid the common problem of over-segmenting text. Segmentation is the first step in a discourse parser, a system that constructs discourse trees from elementary discourse units. We compare SLSeg to a probabilistic segmenter, showing that a conservative approach increases precision at the expense of recall, while retaining a high F-score across both formal and informal texts.

1 Introduction*

Discourse segmentation is the process of decomposing discourse into elementary discourse units (EDUs), which may be simple sentences or clauses in a complex sentence, and from which discourse trees are constructed. In this sense, we are performing low-level discourse segmentation, as opposed to segmenting text into chunks or topics (e.g., Passonneau and Litman (1997)). Since segmentation is the first stage of discourse parsing, quality discourse segments are critical to building quality discourse representations (Soricut and Marcu, 2003). Our objective is to construct a discourse segmenter that is robust in handling both formal (newswire) and informal (online reviews) texts, while minimizing the insertion of incorrect discourse boundaries. Robustness is achieved by constructing discourse segments in a principled way using syntactic and lexical information.

Our approach employs a set of rules for inserting segment boundaries based on the syntax of each sentence. The segment boundaries are then further refined by using lexical information that takes into consideration lexical cues, including multi-word expressions. We also identify clauses that are parsed as discourse segments, but are not in fact independent discourse units, and join them to the matrix clause.

Most parsers can break down a sentence into constituent clauses, approaching the type of output that we need as input to a discourse parser. The segments produced by a parser, however, are too fine-grained for discourse purposes, breaking off complement and other clauses that are not in a discourse relation to any other segment. For this reason, we have implemented our own segmenter, utilizing the output of a standard parser. The purpose of this paper is to describe our syntactic and lexical-based segmenter (SLSeg), demonstrate its performance against state-of-the-art systems, and make it available to the wider community.

2 Related Work

Soricut and Marcu (2003) construct a statistical discourse segmenter as part of their sentence-level discourse parser (SPADE), the only implementation available for our comparison. SPADE is trained on the RST Discourse Treebank (Carlson et al., 2002). The probabilities for segment boundary insertion are learned using lexical and syntactic features. Subba and Di Eugenio (2007) use neural networks trained on RST-DT for discourse segmentation. They obtain an F-score of 84.41% (86.07% using a perfect parse), whereas SPADE achieved 83.1% and 84.7% respectively.

Thanh et al. (2004) construct a rule-based segmenter, employing manually annotated parses from the Penn Treebank. Our approach is conceptually similar, but we are only concerned with established discourse relations, i.e., we avoid potential *same-unit* relations by preserving NP constituency.

*This work was supported by an NSERC Discovery Grant (261104-2008) to Maite Taboada. We thank Angela Cooper and Morgan Mameni for their help with the reliability study.

Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 77–80,
Suntec, Singapore, 4 August 2009.

3 Principles For Discourse Segmentation

Our primary concern is to capture interesting discourse relations, rather than all possible relations, i.e., capturing more specific relations such as Condition, Evidence or Purpose, rather than more general and less informative relations such as Elaboration or Joint, as defined in Rhetorical Structure Theory (Mann and Thompson, 1988). By having a stricter definition of an elementary discourse unit (EDU), this approach increases precision at the expense of recall.

Grammatical units that are candidates for discourse segments are clauses and sentences. Our basic principles for discourse segmentation follow the proposals in RST as to what a minimal unit of text is. Many of our differences with Carlson and Marcu (2001), who defined EDUs for the RST Discourse Treebank (Carlson et al., 2002), are due to the fact that we adhere closer to the original RST proposals (Mann and Thompson, 1988), which defined as 'spans' adjunct clauses, rather than complement (subject and object) clauses. In particular, we propose that complements of attributive and cognitive verbs (*He said (that)...*, *I think (that)...*) are not EDUs. We preserve consistency by not breaking at direct speech (*"X," he said.*). Reported and direct speech are certainly important in discourse (Prasad et al., 2006); we do not believe, however, that they enter discourse relations of the type that RST attempts to capture.

In general, adjunct, but not complement clauses are discourse units. We require all discourse segments to contain a verb. Whenever a discourse boundary is inserted, the two newly created segments must each contain a verb. We segment coordinated clauses (but not coordinated VPs), adjunct clauses with either finite or non-finite verbs, and non-restrictive relative clauses (marked by commas). In all cases, the choice is motivated by whether a discourse relation could hold between the resulting segments.

4 Implementation

The core of the implementation involves the construction of 12 syntactically-based segmentation rules, along with a few lexical rules involving a list of stop phrases, discourse cue phrases and word-level parts of speech (POS) tags. First, paragraph boundaries and sentence boundaries using NIST's sentence segmenter[1] are inserted. Second, a statistical parser applies POS tags and the sentence's syntactic tree is constructed. Our syntactic rules are executed at this stage. Finally, lexical rules, as well as rules that consider the parts-of-speech for individual words, are applied. Segment boundaries are removed from phrases with a syntactic structure resembling independent clauses that actually are used idiomatically, such as *as it stands* or *if you will*. A list of phrasal discourse cues (e.g., *as soon as, in order to*) are used to insert boundaries not derivable from the parser's output (phrases that begin with *in order to...* are tagged as PP rather than SBAR). Segmentation is also performed within parentheticals (marked by parentheses or hyphens).

5 Data and Evaluation

5.1 Data

The gold standard test set consists of 9 human-annotated texts. The 9 documents include 3 texts from the RST literature[2], 3 online product reviews from Epinions.com, and 3 Wall Street Journal articles taken from the Penn Treebank. The texts average 21.2 sentences, with the longest text having 43 sentences and the shortest having 6 sentences, for a total of 191 sentences and 340 discourse segments in the 9 gold-standard texts.

The texts were segmented by one of the authors following guidelines that were established from the project's beginning and was used as the gold standard. The annotator was not directly involved in the coding of the segmenter. To ensure the guidelines followed clear and sound principles, a reliability study was performed. The guidelines were given to two annotators, both graduate students in Linguistics, that had no direct knowledge of the project. They were asked to segment the 9 texts used in the evaluation.

Inter-annotator agreement across all three annotators using Kappa was .85, showing a high level of agreement. Using F-score, average agreement of the two annotators against the gold standard was also high at .86. The few disagreements were primarily due to a lack of full understanding of the guidelines (e.g., the guidelines specify to break adjunct clauses when they contain a verb, but one of the annotators segmented prepositional phrases

[1] http://duc.nist.gov/duc2004/software/duc2003.breakSent.tar.gz

[2] Available from the RST website http://www.sfu.ca/rst/

	Epinions			Treebank			Original RST			Combined Total		
System	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
Baseline	.22	.70	.33	.27	.89	.41	.26	**.90**	.41	.25	**.80**	.38
SPADE (coarse)	.59	.66	.63	.63	**1.0**	.77	.64	.76	.69	.61	.79	.69
SPADE (original)	.36	.67	.46	.37	**1.0**	.54	.38	.76	.50	.37	.77	.50
Sundance	.54	.56	.55	.53	.67	.59	.71	.47	.57	.56	.58	.57
SLSeg (Charniak)	**.97**	.66	**.79**	**.89**	.86	**.87**	**.94**	.76	**.84**	**.93**	.74	**.83**
SLSeg (Stanford)	.82	**.74**	.77	.82	.86	.84	.88	.71	.79	.83	.77	.80

Table 1: Comparison of segmenters

that had a similar function to a full clause). With high inter-annotator agreement (and with any disagreements and errors resolved), we proceeded to use the co-author's segmentations as the gold standard.

5.2 Evaluation

The evaluation uses standard precision, recall and F-score to compute correctly inserted segment boundaries (we do not consider sentence boundaries since that would inflate the scores). Precision is the number of boundaries in agreement with the gold standard. Recall is the total number of boundaries correct in the system's output divided by the number of total boundaries in the gold standard.

We compare the output of SLSeg to SPADE. Since SPADE is trained on RST-DT, it inserts segment boundaries that are different from what our annotation guidelines prescribe. To provide a fair comparison, we implement a coarse version of SPADE where segment boundaries prescribed by the RST-DT guidelines, but not part of our segmentation guidelines, are manually removed. This version leads to increased precision while maintaining identical recall, thus improving F-score.

In addition to SPADE, we also used the Sundance parser (Riloff and Phillips, 2004) in our evaluation. Sundance is a shallow parser which provides clause segmentation on top of a basic word-tagging and phrase-chunking system. Since Sundance clauses are also too fine-grained for our purposes, we use a few simple rules to collapse clauses that are unlikely to meet our definition of EDU. The baseline segmenter in Table 1 inserts segment boundaries before and after all instances of S, SBAR, SQ, SINV, SBARQ from the syntactic parse (text spans that represent full clauses able to stand alone as sentential units). Finally, two parsers are compared for their effect on segmentation quality: Charniak (Charniak, 2000) and Stanford (Klein and Manning, 2003).

5.3 Qualitative Comparison

Comparing the outputs of SLSeg and SPADE on the Epinions.com texts illustrates key differences between the two approaches.

> [Luckily we bought the extended protection plans from Lowe's,] # [so we are waiting] [for Whirlpool to decide] [if they want to do the costly repair] [or provide us with a new machine].

In this example, SLSeg inserts a single boundary (#) before the word *so*, whereas SPADE inserts four boundaries (indicated by square brackets). Our breaks err on the side of preserving semantic coherence, e.g., the segment *for Whirlpool to decide* depends crucially on the adjacent segments for its meaning. In our opinion, the relations between these segments are properly the domain of a semantic, but not a discourse, parser. A clearer example that illustrates the pitfalls of fine-grained discourse segmenting is shown in the following output from SPADE:

> [The thing] [that caught my attention was the fact] [that these fantasy novels were marketed...]

Because the segments are a restrictive relative clause and a complement clause, respectively, SLSeg does not insert any segment boundaries.

6 Results

Results are shown in Table 1. The combined informal and formal texts show SLSeg (using Charniak's parser) with high precision; however, our overall recall was lower than both SPADE and the baseline. The performance of SLSeg on the informal and formal texts is similar to our perfor-

mance overall: high precision, nearly identical recall. Our system outperforms all the other systems in both precision and F-score, confirming our hypothesis that adapting an existing system would not provide the high-quality discourse segments we require.

The results of using the Stanford parser as an alternative to the Charniak parser show that the performance of our system is parser-independent. High F-score in the Treebank data can be attributed to the parsers having been trained on Treebank. Since SPADE also utilizes the Charniak parser, the results are comparable.

Additionally, we compared SLSeg and SPADE to the original RST segmentations of the three RST texts taken from RST literature. Performance was similar to that of our own annotations, with SLSeg achieving an F-score of .79, and SPADE attaining .38. This demonstrates that our approach to segmentation is more consistent with the original RST guidelines.

7 Discussion

We have shown that SLSeg, a conservative rule-based segmenter that inserts fewer discourse boundaries, leads to higher precision compared to a statistical segmenter. This higher precision does not come at the expense of a significant loss in recall, as evidenced by a higher F-score. Unlike statistical parsers, our system requires no training when porting to a new domain.

All software and data are available[3]. The discourse-related data includes: a list of clause-like phrases that are in fact discourse markers (e.g., *if you will, mind you*); a list of verbs used in *to*-infinitival and *if* complement clauses that should not be treated as separate discourse segments (e.g., *decide* in *I decided to leave the car at home*); a list of unambiguous lexical cues for segment boundary insertion; and a list of attributive/cognitive verbs (e.g., *think, said*) used to prevent segmentation of floating attributive clauses.

Future work involves studying the robustness of our discourse segments on other corpora, such as formal texts from the medical domain and other informal texts. Also to be investigated is a quantitative study of the effects of high-precision/low-recall vs. low-precision/high-recall segmenters on the construction of discourse trees. Besides its use in automatic discourse parsing, the system could assist manual annotators by providing a set of discourse segments as starting point for manual annotation of discourse relations.

[3] http://www.sfu.ca/~mtaboada/research/SLSeg.html

References

Lynn Carlson and Daniel Marcu. 2001. *Discourse Tagging Reference Manual*. ISI Technical Report ISI-TR-545.

Lynn Carlson, Daniel Marcu and Mary E. Okurowski. 2002. *RST Discourse Treebank*. Philadelphia, PA: Linguistic Data Consortium.

Eugene Charniak. 2000. A Maximum-Entropy Inspired Parser. *Proc. of NAACL*, pp. 132–139. Seattle, WA.

Barbara J. Grosz and Candace L. Sidner. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12:175–204.

Dan Klein and Christopher D. Manning. 2003. Fast Exact Inference with a Factored Model for Natural Language Parsing. *Advances in NIPS 15 (NIPS 2002)*, Cambridge, MA: MIT Press, pp. 3–10.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8:243–281.

Daniel Marcu. 2000. *The Theory and Practice of Discourse Parsing and Summarization*. MIT Press, Cambridge, MA.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse Segmentation by Human and Automated Means. *Computational Linguistics*, 23(1):103–139.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Aravind Joshi and Bonnie Webber. 2006. Attribution and its Annotation in the Penn Discourse TreeBank. *Traitement Automatique des Langues*, 47(2):43–63.

Ellen Riloff and William Phillips. 2004. *An Introduction to the Sundance and AutoSlog Systems*. University of Utah Technical Report #UUCS-04-015.

Radu Soricut and Daniel Marcu. 2003. Sentence Level Discourse Parsing Using Syntactic and Lexical Information. *Proc. of HLT-NAACL*, pp. 149–156. Edmonton, Canada.

Rajen Subba and Barbara Di Eugenio. 2007. Automatic Discourse Segmentation Using Neural Networks. *Proc. of the 11th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 189–190. Rovereto, Italy.

Huong Le Thanh, Geetha Abeysinghe, and Christian Huyck. 2004. Automated Discourse Segmentation by Syntactic Information and Cue Phrases. *Proc. of IASTED*. Innsbruck, Austria.