A Review of Modern Discourse Parsing Models

Ritvik Rao

Abstract

Over the years, advances in the field of natural language processing (NLP) have allowed for the development of text generation algorithms, virtual assistants such as Google Assistant and Alexa, and machine translation such as Google Translate, among other technologies. However, discourse parsing is one task that has largely stumped researchers. A model with the ability to analyze the structure of an idea or argument as opposed to simply grammar can show insight into how humans formulate thoughts and how ideas can change from person to person, or even from culture to culture. This paper aims to analyze two approaches to the task of discourse parsing: one that involves sentence-level analysis and another that involves text-level analysis. This paper also explores the future of discourse parsing and more advanced technologies that are being considered for further study.

Introduction

Discourse is defined as a discussion about a particular subject, and in NLP, discourse analysis is the subject area that attempts to analyze discourse (Marcu 2000). Discourse parsing is a subtopic within the larger field of discourse analysis. The goal of discourse parsing is to identify the structure of discourse within a given text so that it can be used by another model for further analysis. This is comparable to the building of syntax trees in a compiler to allow another stage in the compiler to make optimizations.

Discourse parsing is important because it can build a model of text that can accurately differentiate between meaningful words and grammatically accurate nonsense (Marcu 2000). A well-designed model can answer pertinent questions about language, such as what the possible relations between two units of texts are and whether such relations are based on the time and place the text is describing or more general principles that transcend settings, what the general structure of text is (treelike? something else?), and if there is any correlation between text structure and perceived importance. Furthermore, discourse parsers can be used to implement further tasks such as summarization, fact checking, and question answering (Marcu 2000).

Despite the potential opportunities and advantages of discourse analysis, the task of discourse parsing is under-explored and considered to be quite difficult. One problem is whether or not discourse can be analyzed at a text level and if there are any patterns worth considering at more than a sentence level. The biggest obstacle is the lack of a unified model (such as the Standard model in physics) for text representation that potential training and testing datasets might follow (Marcu 2000). What follows is a review of a popular text representation paradigm that serves as the base for the two discourse parsing models in this paper.

Rhetorical Structure Theory

Rhetorical Structure Theory (RST) is based on the idea that any text can be broken down into a binary tree that represents how different parts of the text relate to each other with regards to discourse (Mann and Thompson, 1987). RST appears to be the predominant theory that underlines discourse analysis research, and most critics of RST seek to reform rather than replace the theory. The main assumption in RST is that the training text is coherent; it is up to humans to review text and ensure that it makes sense before processing the text with an RST model. There are a number of classes that correspond to types of discourse, such as “Attribution,” “Enablement,” and “Circumstance.” Both models analyzed in

this paper use 18 such classes, since those were the classes found in the utilized corpuses. Along with the coherence assumption, another concept central to RST is rhetorical relation (Mann and Thompson, 1987). Rhetorical relation is the idea that two, non-overlapping sections of text called elementary discourse units (edus) comprise one of the 18 aforementioned classes in discourse. One of these sections is called the "nucleus" and the other is called the "satellite." The nucleus is considered more important to the purpose of the writer because their message does not convey without it; the satellite adds some substance to the discourse but is not required (Mann and Thompson, 1987).

Since RST assumes that all relations have exactly two parts, a given text can be represented as a binary tree that contains all of these relations. For example, the tree below represents the discourse structure tree of the sentence "The bank also says it will use its network to channel investments:"

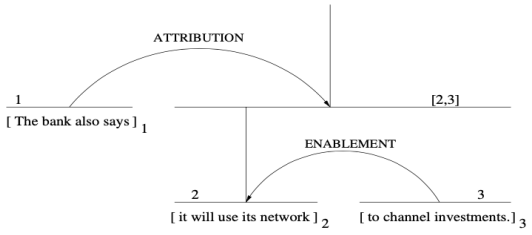

Figure 1: Example discourse parse tree (Soricut and Marcu, 2003)

In this diagram, an arrow is drawn from the satellite edu to the corresponding nucleus. The first relation is attribution, which relates the action of the bank's statement to the statement itself. The diagram implies that the substance of the statement is more essential to simply stating that the bank made this statement. This statement contains its own relation: an enablement relation that connects the use of the network to the intent, "to channel investments." In this case, the action is the nucleus while the justification is the satellite.

There are some drawbacks to RST. The main problem is that there is a continuity assumption: all of the relations do not overlap and there are no gaps (Marcu 2000). This means that RST trees can leave out information if a new relation class arises that the model cannot identify, resulting in gaps in text. Additionally, the theory does not adequately handle text that includes both formal and informal writing, or text that includes a lot of sarcasm.

RST can be applied at two levels. One level is the sentence level, in which all sentence boundaries are marked as class boundaries, which results in discourse analysis at no broader than the sentence level. Another method is text analysis, where rhetorical relations extend throughout a text across sentences and sentence boundaries are simply features to be considered in the model. This literature review seeks to compare and contrast two models, one for each of the paradigms described above.

Sentence level parsing

A team from USC (Soricut and Marcu, 2003) developed a method to build RST-based sentence-level discourse parse trees, such as the one constructed in Figure 1. Their work involved the construction of two models: one model to segment the text into edus and the other to build a probabilistic representation of the rhetoric relation of those edus. Put together, these can be considered as two parts of one model.

Corpuses

To train and test the model, the researchers used 385 Wall Street Journal articles from the Penn Treebank corpus (347 to train and 38 to test). The Penn Treebank corpus appears frequently in discourse analysis research because it is one of the few corpuses that contains text that has been annotated with discourse trees, especially trees in RST format. 53 of the articles also

contain human-annotated trees. The authors use these articles as a point of comparison: human-annotated trees are highly accurate but not perfect, so it is worth treating the accuracy of these humans against the accuracy of the developed model (Soricut and Marcu, 2003). Along with the discourse trees, each sentence has an associated syntax tree, which is a binary tree that uses part-of-speech (POS) and POS tagging to construct relations between different words in a sentence. Preprocessing of the corpus results in 5,809 usable sentences for the model, each of which is represented by the following 3-tuple:

$$\langle s, syntacticTree(s), discourseTree(s) \rangle$$

Figure 2: 3-tuple input into models (Soricut and Marcu, 2003)

In the tuple, "s" refers to a sentence. The syntactic and discourse trees for the training data are used to build the model and the trees for the testing data are used to measure the accuracy of the model.

Discourse Segmentation Model

The first task the researchers needed to complete was to process the text into a tree structure that would simplify the task of discourse parsing. The inputs are the sentences and their lexicalized syntax trees, and the outputs are known as Discourse Segmented Lexicalized Syntax Trees (DS-LSTs), which are simply the syntax trees with the boundaries of the edus in the sentence marked (Soricut and Marcu, 2003). The segmentation model calculates successive probabilities P(b|w,t), where b is one of {boundary, no boundary}, w is a word in a sentence, and t is the syntactic tree for the sentence, derived from either the Penn Treebank or a third-party parser that the researchers reference. Both sources of syntactic trees are used in the researchers' experiments. The formula to calculate all of these probabilities is:

$$P(b|w,t) \simeq \frac{Cnt(N_p \rightarrow \ldots N_w \uparrow N_r \ldots)}{Cnt(N_p \rightarrow \ldots N_w N_r \ldots)}$$

Figure 3: Formula for segmentation

The numerator in the formula is the count of the known edu boundaries at locations between a target word node (N_w) and its sibling node (N_r) at a qualifying parent node (N_p) in a syntax tree t, and the denominator is the number of word node-sibling node pairs in total. This algorithm is called lexical head projection, and the authors used it because they believed it was the best way to use both lexical and syntactic features to calculate the probability that an edu boundary has to be inserted at a certain location. One example of a resulting DS-LST is below:

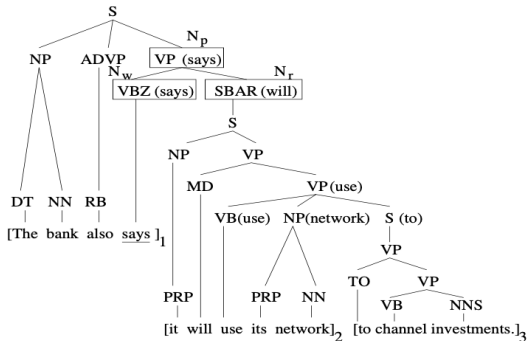

Figure 4: example DS-LST (Soricut and Marcu, 2003)

Ultimately, if P(b|w,t)>0.5, a edu boundary will be added to the tree.

Discourse Parsing Model

The second and main task is to parse the DS-LST trees from the segmentation model. The DS-LST trees demarcate the edus in a sentence, but they do not give any information about how the edus relate to each other (Soricut and Marcu, 2003). The parsing model assigns each adjacent edu pair to a rhetorical relation (choosing the class of the relation and which object is the nuclear edu and which is the satellite) and returns the assignments in a discourse parse tree format. This assignment occurs by cycling through different discourse parse trees by using a dynamic programming algorithm and calculating the likelihood that the given parse

tree reflects the correct assignment. Low-probability trees are sequentially discarded before a good tree is chosen. Since the algorithm searches through the space of all possible parse trees, this is considered to be a bottom-up approach. While time consuming, this is still the fastest approach considering the need to test many different tree possibilities (Soricut and Marcu, 2003). However, it is not practical to use an entire DS-LST tree as an input because of the sparseness of many of the features in the DS-LST tree. Therefore, the authors define a "dominance set," also known as a "filter," of a DS-LST tree that produces only the most important features. Once the dominance set is calculated, the final probability formula is as follows:

$$P(DT|D) = \prod_{c \in DT} P_s(ds(c)|filter_s(c,D)) \times P_r(rel(c)|filter_r(c,D)))$$

Figure 5: Parsing probability formula (Soricut and Marcu, 2003)

The probability that a given parse tree DT is correct, given a DS-LST tree D, is the product of probabilities of a given nuclear-satellite assignment given D and a relation class assignment given D. One example optimal parse tree is shown below:

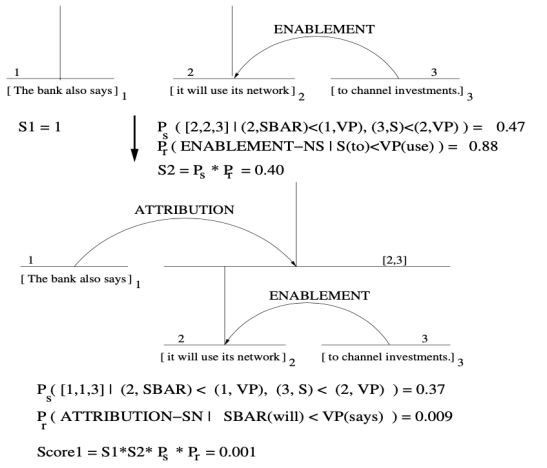

Figure 6: Discourse parse tree (Soricut and Marcu, 2003)

This happens to produce the same tree shown in figure 1.

While Soricut and Marcu provide a robust analysis of sentence-level discourse, most ideas in human language span over longer than one sentence. Therefore, others have preferred to build models that work over more than one sentence.

Text level parsing

A University of Toronto team built a model to identify discourse across sentences (Feng and Hirst, 2012). Unlike the USC researchers who built their model from scratch, the UofT model is based on an existing model that they consider to be the first discourse parser with "state of the art" performance. The UofT model seeks to refine the existing model by choosing a better set of features that results in better performance.

Corpuses

One of the corpuses is the exact set of Wall Street Journal articles used by Soricut and Marcu. The other corpus is a superset of the WSJ articles that are not annotated with RST but with a Penn Discourse Treebank (PDTB) model. Rather than relying on a tree structure, the PDTB model uses a predicate-argument paradigm where "discourse connective" words such as "because" are words that connect two sections of adjacent text (not distinguished by any nuclear/satellite relationship) to create a discourse unit (Feng and Hirst, 2012). This is analogous to a math equation: "because" is like a math operation and the two sections of adjacent text are like input arguments into the equation.

Method

The primary basis for the UofT model was created by a team at the University of Tokyo (Hernault et al., 2010). Their model, called "HILDA," was designed to be a multi-sentence version of the Soricut and Marcu model and relies on a similar process of first building a syntax tree and then classifying the

relations in that tree. There are two major changes that HILDA makes compared to the USC model (Hernault et al., 2010). The first was that the probability of an edu ending at a sentence boundary was no longer 1 because edus could now span across sentences. The second is that there needed to be some way to both combine the syntax trees from different sentences and account for possible gaps in discourse. Therefore, before running the multi-class "relation" classifier, HILDA implements a binary "structure" classifier that decides whether two edus or subtrees of edus should be combined at all or should be left separate (Hernault et al., 2010). This solves the problem of connecting multiple sentences together while accounting for gaps in discourse between sentences, but not necessarily within sentences.

The secondary basis was the work from Lin et al. (2009) that worked on the PDTB database to find the features that maximized the accuracy of their model. Since Lin et al. did not use the RST structure, their entire model could not be used by UofT, but they borrowed the idea that features can be selected before classification is carried out.

Based on the two papers described above, the UofT researchers used the same procedure as the HILDA model, but added many more features. Along with the features the HILDA model used (the same features as the USC model), they use the following categories of features (Feng and Hirst, 2012):

- Lin et al.'s features: includes most of the features from the Lin paper, including dependency parse features and syntactic production rules
- Contextual features: Use the discourse relations of the previous and/or next edu pair as features
- Discourse production rules: see if there is a pattern to what discourse rules appear at each level of a discourse parse tree (e.g., if "Attribute" is the class at the highest node of the parse tree, is there another class that commonly appears at one of the child nodes)
- Semantic similarities: Look at the nouns and verbs in each pair of edus and see if these words commonly appear near each other
- Cue phrases: a concept from an external paper, the UofT researchers look for these phrases in each edu pair

The issue with having all of these categories of features is that it will lead to feature explosion and overfitting, and that so many features will lead to a sparseness problem. Therefore, they borrow the method that the Lin paper uses to reduce features: compare the appearance pattern of each and every feature with the patterns of each RST class, and pick the features that appear with the most RST classes. This will help get rid of very rare features. In all, the UofT team used the 21,410 most common features in their model (Feng and Hirst, 2012).

Results

For both the USC and UofT models, precision and recall are equally important. It is useful to minimize false positives (precision) as well as identify more positive examples accurately than negative ones (recall). Therefore, both models primarily use f-scores, specifically f1-scores, as the metric for model performance. The f1 score is defined as follows:

$$F_1 = \frac{2 * P * R}{P + R}$$

P is equal to precision and R is equal to recall. A higher f1-score is desirable, and the maximum possible score is 100 when representing the value as a percentage.

Since the USC model developed both a discourse segmenter and discourse parser, they measured the performance of both models. Here are the results from the segmenter evaluation:

	Recall	Precision	F-score
$B1DS$	28.2	37.1	32.0
$B2DS$	25.4	64.9	36.5
$DecDS$	77.1	83.3	80.1
$SynDS(T^-)$	**82.7**	**83.5**	**83.1**
$SynDS(T^+)$	**85.4**	**84.1**	**84.7**
HDS	98.2	98.5	98.3

Figure 7: Segmenter evaluation (Soricut and Marcu, 2003)

B1DS and B2DS are naive baseline models that serve as control data (Soricut and Marcu, 2003). DecDS is a segmenter proposed by Marcu in his 2000 book on discourse parsing and is much simpler than the one in the 2003 paper. SynDS(T-) and SynDS(T+) are the discourse segmenters in the USC model; T- uses syntax tree inputs generated by an externally developed algorithm, and T+ uses syntax trees that came with the annotated WSJ articles and are therefore more accurate. HDS is the performance of humans on this task. The table shows that SynDS beat the naïve and older models but has not quite reached the level of humans yet; furthermore, there is only a slight reduction of error by improving the accuracy of the syntax tree inputs (Soricut and Marcu, 2003).

The parser model's performance was also measured. In a similar fashion, the developed parser (SynDP) was compared to the baseline (BDP), an older model (DecDP), and humans (HDP). They also measured the F-scores of the models over unlabeled data, data that used the 18 relation classes, and data that used the 110 relation class-nuclear and satellite label combinations (Soricut and Marcu, 2003). In addition to this, the SynDP model was also tested by changing the accuracy of the syntax tree inputs (T+ vs. T-) and the segmented DS-LST tree inputs (S+ vs. S-). All of the following F-scores are weighted by the frequency of each label as opposed to weighting each label equally; this results in an increase in each F-score. The results are as follows:

	BDP	$DecDP$	$SynDP$	HDP
Unlabeled	64.0	67.0	**70.5**	92.8
18 Labels	23.4	37.2	**49.0**	77.0
110 Labels	20.7	35.5	**45.6**	71.9

Figure 8: F-scores of discourse parsers (T- S-) (Soricut and Marcu, 2003)

	T^-S^-	T^+S^-	T^-S^+	T^+S^+
Unlabeled	70.5	73.0	92.8	**96.2**
18 Labels	49.0	56.4	63.8	**75.5**
110 Labels	45.6	52.6	59.5	**70.3**

Figure 9:SynDP parser vs. accuracy of input trees (Soricut and Marcu, 2003)

Just as before, the SynDP parser beat all models except for the humans. However, the F-scores of the SynDP parser looks very similar to human parsing when assuming that both the syntax tree and segmented trees are built perfectly. This led the researchers to conclude that they had developed a discourse parser that had human-level performance, with the big caveat that the preprocessing steps had to be perfect (Soricut and Marcu, 2003).

At UofT, the training and testing data used was the exact same as in the USC model, which allows for meaningful comparison between the models. The UofT team tested two versions of their model versus the HILDA model and a baseline model. One version used the full set of features obtained after feature selection, and the other version (labeled "NC" in the results) excluded the contextual features because according to the Lin paper, those had the greatest impact on performance (Feng and Hirst, 2012). Additionally, testing was conducted on three different versions of the same dataset: "within-sentence," which only looks at rhetorical relations in the same sentence, "cross-sentence," which only looks at relations between multiple sentences, and "hybrid," which looks at all relations.

Unlike USC, the UofT model did not develop a segmenter, but they did develop a two-step parser as described earlier. The

results of the first step, the "Structure" step, are below:

	Full	NC	HILDA	Baseline
Within-sentence				
Accuracy	**91.04***	85.17*	83.74	53.15
Precision	**92.71***	85.36*	84.81	53.15
Recall	**90.22***	87.01*	84.55	100.00
F_1	**91.45***	86.18*	84.68	69.41
Train F_1	**97.87***	96.23*	95.42	68.52
Cross-sentence				
Accuracy	87.69	86.68	**89.13**	87.82
Precision	49.60	44.73	**61.90**	–
Recall	**63.95***	39.46*	28.00	0.00
F_1	**55.87***	41.93*	38.56	–
Train F_1	**87.25***	71.93*	49.03	–
Hybrid				
Accuracy	**95.64***	87.03	87.04	77.24
Precision	**94.77***	74.19	79.41	–
Recall	**85.92***	65.98*	58.15	0.00
F_1	**89.51***	69.84*	67.13	–
Train F_1	**93.15***	80.79*	72.09	–

Figure 10: Structure F-scores (Feng and Hirst, 2012)

As seen here, the UofT model significantly improves on the HILDA model in cross-sentence testing, and it also does better when contextual features are added. However, it still is not perfect on cross-sentence testing because the selected features do not work as well on those examples; the authors were stumped on how to fix this.

The second part of the UofT model was the "relation" classifier, which is the discourse parser portion of the model. When calculating performance, the model comparisons are completed using the 18 RST classes but not the nuclear-satellite relations. The results are shown below:

	Full	NC	HILDA	Baseline
Within-sentence				
MAFS	**0.490**	0.485	0.446	–
WAFS	**0.763**	0.762	0.740	–
Acc (%)	78.06	**78.13**	76.42	31.42
TAcc (%)	99.90	**99.93**	99.26	33.38
Cross-sentence				
MAFS	**0.194**	0.184	0.127	–
WAFS	**0.334**	0.329	0.316	–
Acc (%)	**46.83**	46.71	45.69	42.52
TAcc (%)	**78.30**	67.30	57.70	47.79
Hybrid				
MAFS	**0.440**	0.428	0.379	–
WAFS	**0.607**	0.604	0.588	–
Acc (%)	**65.30**	65.12	64.18	35.82
TAcc (%)	**99.96**	99.95	90.11	38.78

Figure 11: F-scores of the relation model (Feng and Hirst, 2012)

This data gives us a way to compare the performance of the UofT and USC models. The WAFS (weight-averaged F-score) for the full UofT model is 0.763, compared to the USC model's 0.490 on 18 label data and 0.755 when assuming perfect inputs. Furthermore, while the UofT model improves on HILDA's accuracy and MAFS (regular F-score) in each trial, it fails to meaningfully improve on the weight-averaged F-score in each instance and it has a very low F-score on cross-sentence relations. This is probably because the features selected do not necessarily translate well to a cross-sentence relation (Feng and Hirst, 2012).

From these results, it is clear that the UofT model greatly improves on the USC model on single-sentence classification (0.763 vs. 0.490), likely due to the higher number of features used. Furthermore, the UofT model is capable of noticing relations across sentences, which the USC model cannot do. Both models are able to equal the performance of humans on single-sentence

tasks, which is extremely impressive given that these models were developed quite early (2003 for USC and 2012 for UofT). The UofT model is still more impressive because of its ability to classify at a text-level, even if it cannot do so very perfectly.

There is still a lot of improvement to be done on cross-sentence discourse parsing and on discourse parsing in general, and since the USC and UofT papers have been released, more advances have been made in discourse parsing.

Further analyses of discourse parsing

Recursive neural networks (RNNs) are commonly used in NLP. It therefore makes sense that the technology will eventually be used to do discourse parsing. One research team uses recursive learning to carry out the task of text-level discourse parsing using the same two-step "structure" and "relation" model used by HILDA and the UofT team (Li et al., 2014). The "structure" and "relation" classifiers are each implemented in their own RNN, where at each step the hidden layer is a tanh transformation of the inputs and the output is a final transformation to get a probability; sigmoid is used for the structure classification and softmax for the relation. The tanh transformations mentioned above are called "distributed vectors" by the authors, and at each step the tanh of the dot product of the previous hidden layer and the next vectorized edu segment is calculated. The inputs to the structure RNN are the edu segments from the segmentation step, and the inputs to the relation RNN are the outputs of the structure RNN (Li et al., 2014).

To check the accuracy of the model, the F-score of the model was compared over unlabeled data (span), class-labeled data (relation), and class-and-nuclearity-labeled data (nuclearity). The results are below:

Approach	Span	Nuclearity	Relation
HILDA	75.3	60.0	46.8
Joty et al.	82.5	68.4	55.7
Feng and Hirst	**85.7**	71.0	58.2
Ji and Eisenstein	82.1	**71.1**	**61.6**
Unified (with feature)	82.0	70.0	57.1
Ours (no feature)	82.4	69.2	56.8
Ours (with feature)	84.0	70.8	58.6
human	88.7	77.7	65.7

Figure 12: F-scores for text-level parsing models (Li et al., 2014)

As seen above, the RNN model was comparable to previous models (such as the Feng and Herst/UofT model), but still falls short of human performance. This shows that RNNs have promise as a future structure, but still need to be improved to match human classification.

Another approach that has been suggested is an unsupervised approach. There is still major debate over if there is any merit in looking for patterns above a sentence level, so an unsupervised approach on a text level can help identify patterns if any exist. One paper attempts to use an unsupervised approach to identify four classes of relations (Marcu and Echihabi, 2002). The paper trains over two corpuses: the RAW corpus, which is 1 billion words in unannotated English, and the BLIPP corpus, which contains about 1.8 million automatically parsed sentences. Classification is then attempted using a Naïve Bayes approach, where the authors gather many example pairs of words for each of the four classes they want to detect, then for a Cartesian product of two spans of text, if two words appear that are identified as a pair that indicates a certain class, that class describes the relation between those two spans of text. If more than one class holds, the argmax of the class probabilities is taken (Marcu and Echihabi, 2002). This method simply uses word pairs as the basis for class, not lexical or syntactic patterns or any other assumed patterns of text. The authors then measure their accuracy between two corpuses of text, and they get the following results:

	CONTRAST	CEV	COND	ELAB	NO-REL-SAME-TEXT	NO-REL-DIFF-TEXTS
CONTRAST	-	87	74	82	64	64
CEV			76	93	75	74
COND				89	69	71
ELAB					76	75
NO-REL-SAME-TEXT						64

Table 3: Performances of classifiers trained on the Raw corpus. The baseline in all cases is 50%.

	CONTRAST	CEV	COND	ELAB	NO-REL-SAME-TEXT	NO-REL-DIFF-TEXTS
CONTRAST	-	62	58	78	64	72
CEV			69	82	64	68
COND				78	63	65
ELAB					78	78
NO-REL-SAME-TEXT						66

Table 4: Performances of classifiers trained on the BLIPP corpus. The baseline in all cases is 50%.

Figure 13: Results from Naive Bayes (Marcu and Echihabi, 2002)

As seen above, the classifier is able to beat the 50 percent baseline on all examples on both corpuses (Marcu and Echihabi, 2002). This does not mean that the classifier was accurate; on the contrary, the performance was not close to human performance. However, because of the performance above the baseline, it seems that there are some rules and patterns that apply over a text rather than at the sentence level, and that further work in identifying these patterns is worthwhile.

Conclusion

In this paper, two different models based on rhetorical structure theory were explored: one at the sentence level and one at the text level. Furthermore, neural network and unsupervised approaches were also discussed.

Based on the data and explorations in this paper, it is clear that many advances have been made in the field of discourse parsing. In particular, sentence-level parsing is so accurate that it is comparable to human performance. With that said, there are still many further improvements that need to be made. It is clear that text-level parsing is a worthwhile endeavor, but the task is still not as accurate as human parsing because the patterns at the text level are different than at the sentence level and are not well understood. Furthermore, there needs to be improvements on segmentation models, because large amounts of data will be needed for future parsing, and humans cannot process the amount of data necessary to do discourse parsing in the future.

The solution is then clear: Curate more corpuses with RST discourse trees and RST-style syntax trees and improve segmentation and text-level parsing. Once these steps are completed, discourse parsing will become accurate enough to be used for deep dives into the exploration of human language and the development of state of the art NLP tools.

Works Cited

Feng, V. W., & Hirst, G. (2012, July). Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 60-68).

Hernault, H., Prendinger, H., du Verle, D. A., & Ishizuka, M. (2010). HILDA: A discourse parser using support vector machine classification. *Dialogue & Discourse*, *1*(3), 1-33.

Li, J., Li, R., & Hovy, E. (2014, October). Recursive deep models for discourse parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 2061-2069).

Mann, W. C., & Thompson, S. A. (1987). *Rhetorical structure theory: A theory of text organization* (pp. 87-190). Los Angeles: University of Southern California, Information Sciences Institute.

Marcu, D. (2000). *The theory and practice of discourse parsing and summarization*. MIT press.

Marcu, D., & Echihabi, A. (2002, July). An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th annual meeting of the association for computational linguistics* (pp. 368-375).

Soricut, R., & Marcu, D. (2003). Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 228-235).