2012

Improving discourse structure identification

Jamie Allison Guidry
Louisiana State University and Agricultural and Mechanical College

IMPROVING DISCOURSE STRUCTURE IDENTIFICATION

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The Interdepartmental Program in
Engineering Science

by
Jamie Guidry
B.S., Louisiana State University, 2010
December 2012

For My Daddy,
Col. Clyde James Guidry
November 5, 1946 – November 24, 2010

Acknowledgments

First, I must express my deepest gratitude to my advisor Dr. Gerald Knapp for his guidance, assistance, and unwavering patience. None of this would have been possible without his involvement. But above all, I would like to thank him for having so much faith in me. It has given me enough strength to last a lifetime.

I would also like to thank Dr. Craig Harvey and Dr. Laura Ikuma for serving as members of my committee.

I would like to recognize Leili Javadpour and Mahdi Khazaeli for the help they gave and the knowledge they shared, and the entire semantic analysis research group at LSU for always having my back.

Thanks also to my boss, Wendy Luedtke, for her encouragement and cooperation. Thanks to my great friends, Andrew Bursavich, Shelby Gamble, and Bailey Matens for the awesome brainstorming sessions and for getting me through the tough times. And thanks to my mother, brothers, and sister for the love, motivation, and emotional support that only they could provide.

Table of Contents

List of Tables

List of Figures

Abstract

Rhetorical Structure Theory (Mann et al. 1988), a popular approach for analyzing discourse coherence, suggests that coherent text can be placed into a hierarchical organization of clauses. Identification of a text's rhetorical structure through automatic discourse analysis is a crucial element for many of today's Natural Language Processing tasks, but no sufficient tool is available. The current state-of -the-art discourse parser, SPADE (Soricut et al. 2003), is limited to parsing discourse within a single sentence. HILDA (Hernault et al. 2010) extends the parsing abilities of SPADE to the document level, but with a decrease in performance.

This study achieved document-level discourse parsing without sacrificing performance. Provided text was already segmented into elementary discourse units, the task of discourse parsing was separated into three steps: structuring, nuclearity labeling, and relation labeling. An algorithm was developed for classifying relation existence, nuclearity, and relation label that improved upon previous methods. New features were explored for all three steps to maintain state-of-the-art performance when parsing at the document-level.

Chapter 1 - Problem Statement

1.1 Background

In the field of linguistics, discourse is a unit of language identified by a coherent collection of statements or sentences. For the discourse to be considered coherent, all sentences must contribute to the meaningfulness of the discourse unit (Jurafsky et al. 2009).

The first task of discourse analysis, known as segmentation, is breaking text into elementary discourse units (EDUs). While there are varying viewpoints on the exact definition, all agree that EDUs are non-overlapping spans of text and should be internally coherent (Cristea et al. 1999).

In the subsequent task referred to as labeling or parsing, the structure of and relationships between these EDUs are defined (Jurafsky et al. 2009, p.682). While there are different schools of thought on how to define such relationships, the most widely subscribed to is Rhetorical Structure Theory (RST), a theory about the organization of discourse within written text proposed by Mann et al. (1988) which defines rules for coherent discourse relations. Relations typically consist of two EDUs, a nucleus and a satellite, where the meaning of the satellite is dependent on the nucleus (Jurafsky et al. 2009, p.691). They may exist on the sub-sentence level (between two clauses within a single sentence), the sentence level (between two sentences), and the document level (between groups of sentences or entire paragraphs). These relations result in a hierarchic structure of the text.

Discourse structure has been used for automated text generation since the early 1990s (Rosner et al. 1992; Hovy 1994). It is also used as a tool in automatically evaluating text quality, such as automated essay scoring tools. The Educational Testing Service (ETS) relies on advanced natural language processing technology to automate the scoring of answers to free-response questions. ETS currently uses e-rater, an essay scoring application, in conjunction with a human grader to evaluate essay responses on what are considered high-stakes assessments like the TOEFL and GRE (ETS 2011). E-rater implements automated discourse analysis to evaluate the organization and development of an essay response (Attali et al. 2006). Evaluation of text quality is motivated by the idea that the ordering of information within a discourse affects its coherence (Mann et al. 1988). Lin et al. (2011) used the sequence of relations within a span of text to classify the span as either coherent or incoherent. Other applications of discourse analysis include text summarization (Marcu 1997; Marcu 2000; Louis et al. 2010), content extraction (Taboada et al. 2004; Louis et al. 2010), knowledge extraction (Marir et al. 2002) and argumentation analysis (Mochales et al. 2011).

1.2 Problem Statement

Automated discourse analysis consists of discourse segmentation – breaking documents into elementary discourse units (EDUs) of non-overlapping text spans that are internally coherent – followed by labeling (or parsing) EDUs according to purpose and relationships to other EDUs. There are well-performing tools for discourse segmentation with accuracies of 83% (Soricut et al. 2003; Tofiloski et al. 2009), 86% (Sagae 2009), and 95% (Hernault et al. 2010). Parsing tools for labeling discourse relations are not yet as advanced, the highest accuracies being 63% (Soricut et al. 2003) and 55.3% (Hernault et al. 2010). In their 2003 study, Soricut and Marcu used a feature representation of the syntactic and lexical

information found in the nodes of the parse tree where two discourse segments are joined (which they termed the "dominance set") to estimate the most likely discourse structure of the sentence.

For automated discourse analysis to be further integrated into student essay scoring and other text mining applications, a reliable tool (> 90% accuracy range) is imperative. The 2003 study by Soricut and Marcu revealed a strong connection between lexical syntax and discourse structure, which however was not fully developed in their classifier. This research explores the connection further by developing an additional feature set that incorporates the relations defined by rhetorical structure theory. Moreover, to support its potential use in evaluating student essays, the tool extends on previous methods to identify relations across larger blocks of text up to the document level.

1.3 Objectives

The objectives of this research are as follows:

- Identify features for determining if a relation exists between two spans of text, as well as the nuclearity and type of each relation.
- Develop an algorithm for classifying relation existence, nuclearity, and relation label that improves upon previous methods.
- Implement the model in an application for identifying discourse structure of text.
- Analyze model performance versus the existing SPADE model.

Chapter 2 - Literature Review

2.1 Discourse Parsing

Machine learning is the most widely used approach for automated discourse segmentation and parsing. It was first attempted by Daniel Marcu. In his book *The Theory and Practice of Discourse Parsing and Summarization* (2000), Marcu employed a shift-reduce parsing algorithm with decision rules derived automatically through machine learning to determine the discourse structure of multi-sentence texts.

In a later study, Marcu and Echihabi (2002) used an unsupervised learning approach to label the relation between two discourse units as being one of four groups: contrast, cause-explanation, elaboration, or condition. Two corpora were combined: one of non-annotated English sentences, the other of sentences parsed using Charniak's parser. They achieved 75% - 93% accuracies without relying on discourse cue phrases by using only lexical patterns and span polarity for features.

Reitter (2003) proposed a supervised learning method for rhetorical relation analysis. Although his approach has not been implemented, he provides an in-depth study of the relevant features for training support vector machine classifiers on labeling discourse relations between spans in multi-sentence texts.

SPADE is a tool that performs sentence-level discourse segmentation and parsing through two probabilistic models (Soricut et al. 2003). The model for segmentation assigns a probability of being a discourse boundary to each word in a sentence using lexicalized syntactic parse trees each sentence as features; it yielded an F-score of 83.1%. The parsing model assigns to each possible discourse tree structure the probability that it is the best fit for a particular sentence. Rather than using the syntactic structure of the entire sentence, Soricut and Marcu used the "dominance set" of the sentence. The dominance set contains the word and part of speech at each node of the syntactic parse tree where two EDUs are joined, as well as information about the hierarchy of these nodes within the tree. SPADE labels relations as being one of 18 defined in (Carlson et al. 2001b) with an F-score of 49% when using the author's automatic segmenter and 63.8% when using human-segmented input.

A more recent discourse parser, HILDA (Hernault et al. 2010), uses supervised learning with support vector machines to accomplish Soricut and Marcu's discourse segmentation and relation labeling on the document-level. The authors reported an F-score of 55.3% for HILDA's ability to label identified relations as one of the same 18 classes used in SPADE (Soricut and Marcu, 2003).

Lin et al. (2009) looked specifically into recognizing implicit discourse relations (those with no signal or cue phrase) using a supervised learning approach based on a maximum entropy classifier. Given two spans of text pre-identified as having an implicit relation, their model classifies the relationship as being one of the 11 most prominent second level relation types, as defined by the Penn Discourse Treebank Research Group (2007). Syntactic structure of both text spans, dependency patterns of the previous and following pairs of text, the dependency tree of both spans, and word pairs across spans were the four groups of features used, resulting in 40.2% accuracy. Pitler et al. (2009) extended Lin's study with additional features to represent span polarity, length of a spans verb phrase, modal verbs, and the presence of currency or other numeric values. These features increased performance with an F-score of 47.1%.

Subba et al. (2009) used inductive logic programming to classify two spans of text as having one of 26 relations. Their parser considered the semantic representation using WordNet, linguistic cues like tense and modality, hypernym and meronym relations using WordNet, structural information, and cosine similarity. It performed well when identifying relations within a single sentence with an F-score of 63%, but only achieved an F-score of 35.4% when parsing an entire multi-sentence document. The authors believed the poor performance resulted from errors made when constructing the lower levels of the structure tree.

2.2 Discourse Segmentation and Other Supporting Tools

Other researchers have focused their efforts on developing technology to support discourse parsing. Thanh et al. (2004) used a rule-based algorithm to address discourse segmentation and nuclearity identification. However, the discourse units output by the segmenter lack the semantic coherence needed for proper discourse parsing. Subba et al. (2007) proposed a neural network model for segmenting sentences into EDUs. They achieved an F-score of 84.4%, but as with Thanh et al., the resulting EDUs are not conducive to the parsing goals of this research, as their EDUs do not all contain verb phrases. Suitable segments must contain a verb phrase, a concept supported by Tofiloski et al. (2009). Tofiloski's open source segmenter, SLSeg, uses predefined rules to segment multi-sentence text into elementary discourse units with an F-score of 83%.

Chapter 3 - Methodology

3.1 Overview

The parsing process consists of four tasks (Figure 1): segmentation, structuring, nuclearity labeling, and relation labeling. The first step, segmentation, is handled by an existing segmenter (such as Tofiloski et al. 2009) and is not a focus of this research. The second and third steps (Section 3.3.1) construct the discourse tree for the entire text and determine the nuclearity of each relation, respectively. The fourth and final step (Section 3.3.2) labels each relation in the discourse tree with one of 17 classes. The algorithm used for creating the discourse tree is inspired by the algorithm used by Hernault et al. (2010) in the development of HILDA. Changes proposed in this study aim to make classification more reliable and computationally efficient. HILDA uses a single classifier for relation and nuclearity labeling, performed in the final stage of the parsing process. Determining nuclearity earlier in the process presents the opportunity to refine the features used for classification in later steps.

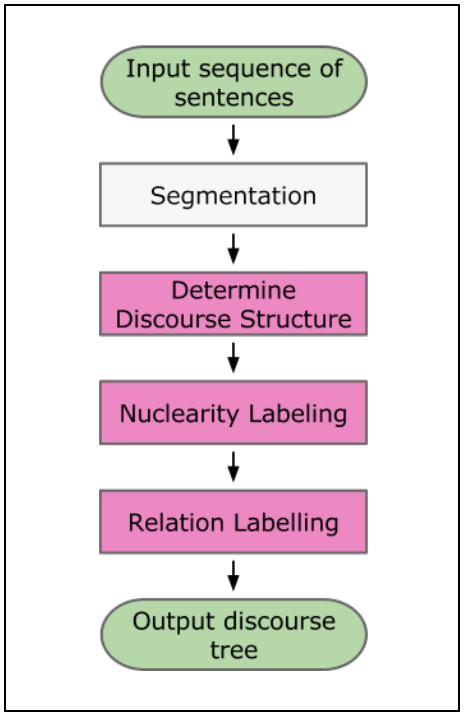

Figure 1. Methodology overview.

The implementation was developed in Windows 7 using the Python programming language to call NLP preprocessing tools, extract features, call classifiers, and store and organize output. Neural network models were trained in MATLAB, and Support Vector Machine (SVM) classifiers were trained in Weka (Hall et al. 2009).

3.2 Corpus

The dataset used for this study is the RST Discourse Treebank, a corpus of over 300 articles from the Wall Street Journal with the discourse structure annotated according to Rhetorical Structure Theory (Carlson et al. 2002). The annotated files include the segmented EDUs in their hierarchical discourse

structure with nuclearity and relationships (defined in Section 3.3.2, Table 4) identified. Following is a sample article from the corpus.

> Example 3.1. "Spencer J. Volk, president and chief operating officer of this consumer and industrial products company, was elected a director. Mr. Volk, 55 years old, succeeds Duncan Dwight, who retired in September." (Carlson et al. 2002)

The discourse tree and corpus annotation for this article are depicted in Figure 2 and Figure 3, respectively.

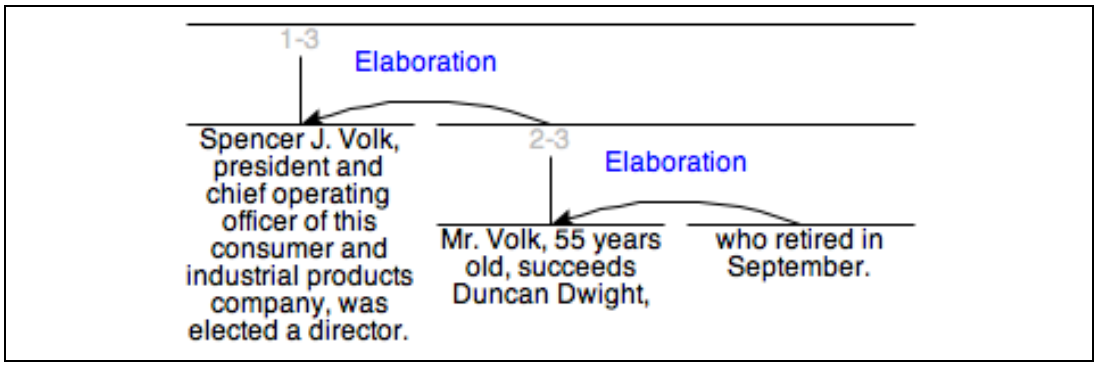

Figure 2. RST discourse tree for Example 3.1.

```
( Root (span 1 3) (prom 1)
    ( Nucleus (leaf 1) (rel2par span) (prom 1) (text Spencer J. Volk, president
      and chief operating officer of this consumer and industrial products
      company, was elected a director.) )
    ( Satellite (span 2 3) (rel2par elaboration-additional) (prom 2)
        ( Nucleus (leaf 2) (rel2par span) (prom 2) (text Mr. Volk, 55 years old,
          succeeds Duncan Dwight,) )
        ( Satellite (leaf 3) (rel2par elaboration-additional-e) (prom 3) (text who retired
          in September.) )
    )
)
```

Figure 3. Annotation for Example 3.1 (Carlson et al. 2002).

3.3 Parsing Process

3.3.1 Structuring and Nuclearity

The hierarchical discourse structure of the input text is formed using a bottom-up parsing approach. This step incorporates a two-phase model. The first phase is a two-layer feed-forward neural network, hereafter referenced as *Rel* (Relation). *Rel* uses the features of two nodes in a discourse tree to estimate the likelihood that a relation exists between the two nodes. The features used are listed below with features unique to this study marked with an asterisk. The term "span" refers to any node in the discourse tree. While this may be an EDU, in some cases it may be a branch of the discourse trees with child EDUs as its leaves.

- **Span length** - number of words in the span
- **Sentence length** - number of words in the parent sentence of the current span
- **Sentence location** - sentence number divided by the total number of sentences
- **Child nodes** - number of EDUs contained in the span
- **Spans before -** number of spans before in same sentence (if span is within a single sentence)
- **Spans after** - number of spans after in same sentence (if span is within a single sentence)
- **Cue phrase presence* -** Boolean if the segment contains at least one of the discourse cue phrases
- **Verb tense* -** tense of verb in span's verb phrase, obtained by parsing each EDU with the Stanford Parser (Klein et al. 2003).

For each article in the corpus, features were extracted for every possible combination of adjacent spans. See Table 1 for the possible combinations of the article in Example 3.1.

Table 1. Potential relations for Example 3.1.

Left Span EDUs	Right Span EDUs	Related?
1	2	False
1	2 – 3	True
1 – 2	3	False
2	3	True

A class value of 1 was appended to the end of the feature vector for combinations that were present in the corpus .dis file, and a zero was appended otherwise. Because this resulted in such a large number of negative examples, Python's "random" library was used obtain equal sample sizes of each class for training.

Following *Rel*'s identification of relation existence, a multi-class Support Vector Machine, *Nuc* (Nuclearity), assigns one of three classes: NS (a mononuclear relation where the left span is the nucleus and the right is the satellite), SN (a mononuclear relation where the right span is the nucleus), and NN (a multinuclear relation where both spans identify as a nucleus). SVM models were trained using Weka and LibSVM (Chang et al. 2011). In addition to the feature set used for the *Rel* model, *Nuc* incorporates several lexical and syntactic features.

One set looks for the presence of certain punctuation and Penn Treebank tags in the leftmost node of each span. The following tags consistently proved useful in determining nuclearity: SBAR (subordinate clause), CC (coordinating conjunction), JJR (comparative adjective) or RBR (comparative adverb), JJS (superlative adjective) or RBS (superlative adverb), MD (modal), LS (list item marker), a comma, and quotation marks. More features are taken from the named entity annotation output from the Stanford CoreNLP (Finkel et al. 2005). The pronoun subject (starred below) is a new feature to this work. The following features are extracted from both the left and the right span:

- **SBAR presence**
- **CC presence**
- **JJR or RBR presence**
- **JJS or RBS presence**
- **MD presence**

- **LS presence**
- **Comma presence**
- **Quotation Mark presence**
- **Temporal (Date or Time) entity presence**
- **Ordinal entity presence**
- **Set entity presence**
- **Duration entity presence**
- **Subject is pronoun*** – subject is tagged as PRP or WP
- **Child relation** – label of relation which formed the span
- **Child nuclearity** – label of relation which formed the span
- **Cue phrase** – cue phrase if present; if more than one is present, the one closest to span boundary

Upon analysis of the corpus, it became evident that nuclearity was often signaled by attributes representative of the combined spans. Certain words types signaled nuclearity when located at the inner boundary of the combined spans. For one of these words to be considered present, any sense or predefined synonym of the word could be present. The lists of words included in each of these groups are shown in Figure 4.

```
words_because = ['because','since','for']
words_say = ['said','say','says','saying','reported','noted','announced','indicated','stated']
words_which = ['which','who','whom','where']
```

Figure 4. Boundary word groups.

Four Boolean features are defined based on these word groups:

- **Final word of left span is in "say" group**
- **First word of right span is in "say" group**
- **First word of right span is in "because" group**
- **First word of right span is in "which" group**

Two final features are used to describe the two spans as a pair:

- **Both spans are in same sentence** – Boolean
- **Pair type** – two leaves (neither has descendants), one leaf (has no descendants) and one span (has descendants), two spans (both have descendants)

Figure 5 summarizes the algorithm for forming the discourse tree of the text. The structuring algorithm takes as input an array *S* of size *N*, where *N* is the total number of EDUs in the text. The *Rel* model is called for every pair of consecutive EDUs, and the pair with the highest likelihood of having a relation then uses the *Nuc* model to determine which of the two serves as the nucleus of the discourse relation. The two EDUs are removed from *S* and replaced with *T* representing the combination of EDUs. Contextual features are recalculated and used, and the cue phrase and verb tense features are inherited from the left span of *T*. This process is repeated until the only item in the list is a single element, *T*, which is the discourse tree for the entire text.

Considering the same sample article from Example 3.1, the array of segmented EDUs is shown in Table 2. The starting values for the structuring features are given in Table 3. Figure 6 shows a complete demonstration of the algorithm on the same article.

Input: S
  $N \leftarrow$ number of items in array S
  while $N > 1$ **do**
    Create an empty array R
    $i \leftarrow 1$
    for $j = 2 \rightarrow N$ **do**
      $\vec{r} \leftarrow \textsc{Rel}(\vec{x}_i, \vec{x}_j)$
      append $\vec{r}$ to R
      $i \leftarrow i + 1$
      $j \leftarrow j + 1$
    end for
    select from R the $\vec{x}_i, \vec{x}_j$ pair with maximum P
    $\textsc{Nuc}(\vec{x}_i, \vec{x}_j)$
    $T \leftarrow i$ & j
    remove $\vec{x}_i$ and $\vec{x}_j$ from S
    append $\vec{x}_T$ to S
    $N \leftarrow$ number of items in array S
  end while
  function $\textsc{Rel}(\vec{x}_i, \vec{x}_j)$
    $P \leftarrow$ probability of relation between spans i and j
    return $< \vec{x}_i, \vec{x}_j, P >$
  end function
  function $\textsc{Nuc}(\vec{x}_i, \vec{x}_j)$
    identify nucleus of i and j
  end function

Figure 5. Algorithm for constructing discourse tree.

Table 2. Array of EDUs for Example 3.1.

1	Spencer J. Volk, president and chief operating officer of this consumer and industrial products company, was elected a director.
2	Mr. Volk, 55 years old, succeeds Duncan Dwight,
3	who retired in September.

Table 3. Starting features for discourse tree formation of Example 3.1.

EDU ID	Span length	Sent. length	Sent. location	Child nodes	Spans before	Spans after	Cue phrase presence	Verb tense
1	19	19	1/2	1	0	2	1	VBD
2	8	12	2/2	1	1	1	0	VBZ
3	4	12	2/2	1	2	0	0	VBD

Initially, the array of feature vectors $S = \begin{Bmatrix} < \mathbf{1}: 19, 19, \frac{1}{2}, 1, 0, 2, 1, \text{VBD} > \\ < \mathbf{2}: 8, 12, \frac{2}{2}, 1, 1, 1, 0, \text{VBZ} > \\ < \mathbf{3}: 4, 12, \frac{2}{2}, 1, 2, 0, 0, \text{VBD} > \end{Bmatrix}$,

the total number of spans $N = 3$

Iteration 1

Suppose the likelihood of spans being related $P(1,2) = 0.1, P(2,3) = 0.7$

$R = \begin{Bmatrix} < \overrightarrow{x_1}, \overrightarrow{x_2}, 0.1 > \\ < \overrightarrow{x_2}, \overrightarrow{x_3}, 0.7 > \end{Bmatrix}$

$\max(P) = 0.7$

$\overrightarrow{x_i} \leftarrow \overrightarrow{x_2}$

$\overrightarrow{x_j} \leftarrow \overrightarrow{x_3}$

Suppose $Nuc(\overrightarrow{x_2}, \overrightarrow{x_3}) = \overrightarrow{x_2}$

$T =$ "Mr. Volk, 55 years old, succeeds Duncan Dwight, who retired in September."

$\overrightarrow{x_T} =< \mathbf{2}: 12, 12, \frac{2}{2}, 2, 1, 0, 0, \text{VBZ} >$

$S = \begin{Bmatrix} < \mathbf{1}: 19, 19, \frac{1}{2}, 1, 0, 2, 1, \text{VBD} > \\ < \mathbf{2}: 12, 12, \frac{2}{2}, 2, 1, 0, 0, \text{VBZ} > \end{Bmatrix}, N = 2$

Iteration 2

Suppose $P(1,2) = 0.3$

$R = \{< \overrightarrow{x_1}, \overrightarrow{x_2}, 0.3 >\}$

$\max(P) = 0.3$

$\overrightarrow{x_i} \leftarrow \overrightarrow{x_1}$

$\overrightarrow{x_j} \leftarrow \overrightarrow{x_2}$

Suppose $Nuc(\overrightarrow{x_1}, \overrightarrow{x_2}) = \overrightarrow{x_1}$

$T =$ "Spencer J. Volk, president and chief operating officer of this consumer and industrial products company, was elected a director. Mr. Volk, 55 years old, succeeds Duncan Dwight, who retired in September."

$\overrightarrow{x_T} =< \mathbf{1}: 31, 19, \frac{1}{2}, 3, 0, 0, 1, \text{VBD} >$

$S = \{< \mathbf{1}: 31, 19, \frac{1}{2}, 3, 0, 0, 1, \text{VBD} >\}$, $\boxed{N = 1}$ STOP

Figure 6. Structuring algorithm example.

3.3.2 Labeling

The corpus uses a total of 78 discourse relations. For this study, the relations are be grouped into 16 relation categories defined by Carlson et al. (2001a). Table 4 lists the 16 categories with the more specific relations that each contains. Each relation in the corpus was replaced with the label of the category under which it falls.

Table 4. Relation groups.

ATTRIBUTION	attribution, attribution-negative
BACKGROUND	background, circumstance
CAUSE	cause, result, consequence
COMPARISON	comparison, preference, analogy, proportion
CONDITION	condition, hypothetical, contingency, otherwise
CONTRAST	contrast, concession, antithesis
ELABORATION	elaboration-additional, elaboration-general-specific, elaboration-part-whole, elaboration-process-step, elaboration-object-attribute, elaboration-set-member, example, definition
ENABLEMENT	purpose, enablement
EVALUATION	evaluation, interpretation, conclusion, comment
EXPLANATION	evidence, explanation-argumentative, reason
JOINT	list, disjunction
MANNER-MEANS	manner, means
TOPIC-COMMENT	problem-solution, question-answer, statement-response, topic-comment, comment-topic, rhetorical-question
SUMMARY	summary, restatement
TEMPORAL	temporal-before, temporal-after, temporal-same-time, sequence, inverted-sequence
TOPIC CHANGE	topic-shift, topic-drift

Two additional labels are seen in the corpus, which assist in structuring the discourse trees: TEXTUAL ORGANIZATION and SAME-UNIT. Although the use of both is supported by Soricut and Marcu (2003) and Hernault et al. (2010), TEXTUAL ORGANIZATION will be disregarded in this study. It is one of the most infrequently seen relations in the corpus. Of the 20,015 relation extracted, there are only 152 instances of TEXTUAL ORGANIZATION. Additionally, the Discourse Tagging Reference Manual defines the TEXTUAL ORGANIZATION relation as "a multinuclear relation used to link elements of the structure of the text, for example, to link a title with the body of the text, a section with the text of a section, etc." (Carlson et al. 2001a). Because the corpus is made of news articles, the TEXTUAL ORGANIZATION label is used to accommodate the headlines, captions, and author listings that appear in the text. Since this relation neither makes up a significant portion of the data nor contributes to the analysis of natural discourse, the 152 samples are thrown out before training any models. The SAME-UNIT label is maintained, however, resulting in a total of 17 relation labels.

Each relation in the newly constructed discourse tree is labeled as one of the relations in Table 4 using a k-nearest neighbor (k-NN) learning algorithm. This will use a bottom-up approach, labeling the discourse tree from the lowest level relation to the highest level in the discourse. The labeling classifier, *Label*, uses the same feature set as *Nuc* (Section 3.3.1).

Chapter 4 – Results and Analysis

All models in this study were evaluated using the F-score performance metric. The formulae for calculation of the F-score are shown in Figure 7. Where relevant, performance is compared to SPADE, the current state of the art discourse parser (Soricut et al. 2003). As SPADE only parses discourse within a single sentence, performance will also be compared to HILDA, a document-level discourse parser (Hernault et al. 2010).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$Fscore = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Figure 7. F-score calculation.

4.1 Relation Detection

From the final dataset of 22,207 samples, 65% were used for training, 15% for validation, and 20% for testing. The samples were divided up randomly into the three groups. Ten-fold cross validation was used to minimize the impact of specific case selections on performance results. The goal of this phase was to identify the existence of relations between spans, building the blank discourse structure of the text from the bottom up.

Prediction of relation existence between spans was performed using MATLAB's Neural Network Toolbox. Results were excellent, with an F-score of 94.8%. The most comparable statistic from Soricut and Marcu's 2003 study is their reported F-score of 92.8% for their parser's ability to construct an unlabeled discourse tree using automatic syntactic parsing and human segmented input. Their parser only identifies discourse structure within a single sentence however. Hernault et al. (2010) achieved an F-score of 83.0% for HILDA, their document-level discourse parser, in the structuring phase of the blank discourse tree using human segmented input.

Table 5. Summary of *Rel* results.

Fold	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Total**
True Positives	2155	2174	2117	2024	2122	2103	2124	2158	2136	2106	21219
True Negatives	2098	2055	2131	2106	2085	2132	2105	2087	2102	2154	21055
False Negatives	68	72	81	166	95	84	68	77	71	67	849
False Positives	120	140	112	145	139	122	127	119	132	114	1270
										Precision:	0.9435
										Recall:	0.9615
										F-Score:	0.9479

4.2 Nuclearity

Given that two spans are determined to be related, the second model, *Nuc*, determines the nuclearity. Nuclearity can be one of three classes: NS (a mononuclear relation where the left span is the nucleus and the right is the satellite), SN (a mononuclear relation where the right span is the nucleus), and NN (a multinuclear relation where both spans identify as a nucleus).

Using a multiclass SVM to classify relations as one of the three nuclearity types yielded an overall F-score of 68.3%. Table 6 shows the performance broken down by class. These scores were obtained through 10-fold cross validation of the full data set. The classes were balanced by enforcing a 5000-sample cap per class.

Table 6. Summary of multiclass *Nuc* results.

Class	F-score	Sample Size
NS	0.713	5000
SN	0.699	3314
NN	0.636	4360

Table 7. Multiclass *Nuc* confusion matrix.

NS	**SN**	**NN**	← classified as
4091	376	533	**NS**
714	2073	527	**SN**
1665	165	2530	**NN**

In order to compare methods, the model was also trained on the standard training set provided in the corpus and evaluated on the standard test set, resulting in an average F-score of 70.8%. Using the same data sets and evaluation method, HILDA performed with an F-score of 68.4%. The results obtained upon cross-validation provide a better idea of how the model will ultimately perform with unknown data.

Features were evaluated using Weka's Information Gain Ranking attribute evaluator. Table 8 lists the highest-ranking features. Table 9 lists the specific cue word the evaluator found to signal nuclearity when present near the right boundary of the left span or left boundary of the right span, respectively. Even though a few of the word group features described in Section 3.3.1 use the first word of the right span, there were other boundary words that improved performance. The beginning of the right span usually gives the most information about the relation since this is where the transition takes place. All right span first words were included as features before reducing the feature set with the attribute evaluator. All words ranking above zero (see Table 10) were used in training the model.

Table 8. Feature ranking for *Nuc*.

Subordinating conjunction - SBAR (R)	0.139	Modal - MD (L)	0.005
Span ends with "say" word (L)	0.137	Subject is pronoun (L)	0.005
Span Length (L)	0.128	Sentence length (L)	0.004
Count of EDUs before (R)	0.127	Span starts with "because" word (R)	0.004
Count of child nodes (L)	0.123	Span starts with "say" word (R)	0.004
Count of EDUs after (L)	0.065	Named entity: DATE or TIME (L)	0.004
Span Length (R)	0.065	Verb is present participle -VBG (L)	0.004
Span has no child nodes (L)	0.055	Comparative adjective or adverb - JJR/RBR (L)	0.004
Both spans in same sentence	0.053		
Both are spans have child nodes	0.041	Verb is 3rd person singular present tense - VBZ (L)	0.004
Subject is pronoun (R)	0.038		
Modal - MD (R)	0.029	Quotation mark (L)	0.004
Span starts with "which" word (R)	0.028	Verb is base form – VB (L)	0.003
Coordinating conjunction - CC (R)	0.027	Sentence location (L)	0.003
Count of child nodes (R)	0.023	Verb is singular present tense - VBP (L)	0.003
Both spans are leaf nodes	0.022	Count of EDUs after (R)	0.003
One span has child nodes, one is leaf node	0.019	Named entity: DURATION (L)	0.002
Verb is past tense - VBD (L)	0.015	Superlative adjective or adverb - JJS/RBS (L)	0.002
Verb is present participle - VBG (R)	0.013		
Verb is past participle - VBN (R)	0.013	Verb is 3rd person singular present tense - VBZ (R)	0.002
Verb is base form -VB (R)	0.011		
Sentence length (R)	0.007	Comparative adjective or adverb - JJR/RBR (R)	0.001
Span has no child nodes (R)	0.006		
Comma (R)	0.006	List item marker - LS (R)	0.001
Quotation mark (R)	0.006		

Table 9. Cue word feature ranking for *Nuc*.

Left Span		Right Span			
if	0.0041	and	0.029	since	0.0015
although	0.0028	none	0.0172	when	0.0015
for	0.0023	but	0.0071	meanwhile	0.0014
and	0.002	also	0.0031	for	0.0013
though	0.002	while	0.003	or	0.0012
but	0.0019	because	0.0028	next	0.0012
however	0.0018	as	0.0021	then	0.0011
or	0.0013	after	0.0016	in fact	0.0005
either	0.0008	still	0.0015	otherwise	0.0005

Table 10. First word of right span feature ranking for *Nuc*.

and	0.0702	would	0.0027	didn't	0.0011	created	0.0005
that	0.0622	its	0.0027	had	0.0011	dollar	0.0005
it	0.0287	ended	0.0025	his	0.001	fell	0.0005
to	0.024	i	0.0023	at	0.001	filed	0.0005
which	0.0171	we	0.0021	net	0.001	hasn't	0.0005
is	0.0119	said	0.0021	meanwhile	0.001	helped	0.0005
they	0.0093	via	0.0019	called	0.0009	include	0.0005
who	0.0093	whose	0.0019	from	0.0009	included	0.0005
but	0.0076	because	0.0019	declined	0.0009	involving	0.0005
was	0.0053	after	0.0018	following	0.0009	living	0.0005
will	0.0053	where	0.0017	others	0.0009	must	0.0005
or	0.0049	by	0.0016	ranging	0.0009	noting	0.0005
in	0.0049	based	0.0016	do	0.0007	otherwise	0.0005
has	0.0044	for	0.0014	isn't	0.0007	partly	0.0005
are	0.0043	also	0.0014	known	0.0007	produced	0.0005
as	0.004	since	0.0014	largely	0.0007	reflecting	0.0005
have	0.0035	of	0.0013	may	0.0007	related	0.0005
according	0.0034	then	0.0013	nor	0.0007	remains	0.0005
while	0.0032	ending	0.0012	showed	0.0007	saw	0.0005
he	0.0032	priced	0.0012	inflation	0.0005	see	0.0005
were	0.003	when	0.0012	much	0.0005	suggests	0.0005
says	0.0028	under	0.0011	apparently	0.0005	us	0.0005
including	0.0028	how	0.0011	backed	0.0005	using	0.0005
whether	0.0028	can	0.0011	commodities	0.0005	wouldn't	0.0005

4.3 Labeling

The *Label* classifier uses a k-NN learning algorithm implemented in MATLAB using the corpus's standard training data set and evaluated using the standard test data set. Setting k = 15 yielded the highest overall F-score of 64.5%. This is an improvement upon both SPADE and HILDA. SPADE, the sentence-level parser, achieved an F-score of 63.8% when classifying 18 labels, using automatic syntactic parsing and human segmented input (Soricut et al. 2003). HILDA had an F-score of 55.3% (Hernault et al. 2010).

Observation of the performance by class (Table 11) revealed that the high F-score resulted from good performance on the classes dominating the data set, while the remaining classes experienced poor results. *Label* achieves acceptable results for ATTRIBUTION, CONDITION, CONTRAST, ELABORAION, JOINT, and SAME-UNIT relation groups. The classifier is unable to identify any TOPIC-COMMENT, TEMPORAL, CAUSE, or EVALUATION samples, and the remaining classes have poor performance.

Soricut et al. (2003) do not provide class-level results, so it is unknown whether their parser performed consistently for all classes or simply favored the more frequent ones. Hernault et al. (2010) do provide performance results by class. The authors state that several of the relation labels are not present in the test of the corpus; among them are TOPIC-COMMENT and EVALUATION. All of the excluded classes have low-performance. The authors do not state whether these classes are present in the

calculation of the overall labeling performance. Neither their source code nor detailed methodology is publicly available, so their efforts cannot be duplicated to validate their results. The classes excluded by Hernault et al. (2010) do in fact exist in the standard test set of the RST Discourse Treebank corpus, though in small number, and will still be considered in this study.

Table 11. Summary of *Label* results.

Class	Sample Size		Precision	Recall	F-Score
	Train	Test			
ATTRIBUTION	2627	302	0.847	0.937	0.890
BACKGROUND	190	102	0.387	0.118	0.180
CAUSE	584	75	0.0	0.0	0.0
CHANGE	187	12	0.143	0.167	0.154
COMPARISON	269	27	0.500	0.037	0.069
CONDITION	258	43	0.800	0.465	0.588
CONTRAST	845	134	0.734	0.351	0.475
ELABORATION	6622	682	0.604	0.952	0.739
ENABLEMENT	475	45	0.385	0.333	0.357
EVALUATION	500	71	0.0	0.0	0.0
EXPLANATION	941	96	0.583	0.146	0.233
JOINT	1628	181	0.587	0.840	0.691
MANNER-MEANS	190	25	0.500	0.040	0.074
SAME-UNIT	1210	117	0.729	0.966	0.831
SUMMARY	177	29	0.333	0.034	0.063
TEMPORAL	426	68	0.0	0.0	0.0
TOPIC-COMMENT	126	22	0.0	0.0	0.0

Because ELABORATION is more than double the size of the next largest class (ATTRIBUTION), the classifier is biased, skewing the results. This makes it difficult to determine where the confusion truly occurs. Column P of the Label confusion matrix (Table 12) illustrates all of the false positives under ELABORATION.

To gain more insight, ELABORATION was eliminated from the data set and a Support Vector Machine classifier, *Label_NoElab* was trained and evaluated using 10-fold cross validation on the entire corpus. The remaining class sizes were balanced using the SpreadSubsample filter in Weka, setting the maximum sample size for any class to 300 instances.

In the absence of ELABORATION, the relation groups CHANGE, ENABLEMENT, and MANNER-MEANS improved to an acceptable level. The remaining groups are addressed below the confusion matrix for *Label_NoElab* (Table 13).

Table 12. Nearest neighbor *Label* confusion matrix.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	← classified as
14	1														81		**A = Explanation**
	47			2	10		29	12		2				1	30	1	**B = Contrast**
		1		1				2							21		**C = Manner-Means**
			15					1							29		**D = Enablement**
	7			12	4		4	13		3	1	1			56	1	**E = Background**
					113		3	1									**F = Same-Unit**
1	1			1			10	1							6	2	**G = Topic-Comment**
			1		15		152	1							5	7	**H = Joint**
				1	5			283							13		**I = Attribution**
		1		10	1		37	1							18		**J = Temporal**
1	1				1			5		20					15		**K = Condition**
4	5		2		1		9	3				1			50		**L = Cause**
			2									1			26		**M = Summary**
1				1	1		1	4							63		**N = Evaluation**
	1		1	1	2		5	2						1	13	1	**O = Comparison**
3			18	2	2			5					3		649		**P = Elaboration**
	1						9									2	**Q = Change**

Table 13. SVM *Label_NoElab* confusion matrix.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	← classified as
114	1	4	2	5			2	4		8	23	129	1	7		**A = Explanation**
13	69	1		26	8		59	14	4	25	8	29	37	7		**B = Contrast**
10		131	14	7				12		7	15	15		4		**C = Manner-Means**
4		11	248	9				9		10	2	7				**D = Enablement**
35	17	12	6	72				28	1	39	17	51		22		**E = Background**
					247		23		24				6			**F = Same-Unit**
5	2			8	4		34		4	4	2	25	60			**G = Topic-Comment**
					25		193		10				72			**H = Joint**
4	1	1	2	11				223		23	15	2		18		**I = Attribution**
14	5	17	6	23	13		104	14	30	16	8		40	10		**J = Temporal**
7	2	2	8	15	3		7	10		198	8	32	1	7		**K = Condition**
43	7	23	28	33	5		29	13	6	12	35	55	7	4		**L = Cause**
41	5	4	8	12	1		2	12		8	7	196	1	3		**M = Evaluation**
2	1				4		9					5	178			**N = Change**
23	9	7	17	10	15		60	12	20	22	16	26	20	39		**O = Comparison**
34		2	62	10				5		1	18	66		4	4	**P = Summary**

Background

Label classified more than half of the BACKGROUND instances as ELABORATION. Removing ELABORATION with *Label_NoElab* only slightly improves classification. The two relations contained in the BACKGROUND class are "background" and "circumstance". "Background" relations are difficult to detect, especially computationally, as they often require knowledge beyond the context of the discourse. In the example below (Figure 8), one would need to know that the release of toxic substances into the environment is frowned upon by the government. The tagging manual even warns that "the information or the context of the background relation is not always specified clearly or delimited sharply." (Carlson et al. 2001a).

```
( Satellite (span 89 90) (rel2par background)
    ( Nucleus (leaf 89) (rel2par span) (text _!But Superfund also contains a criminal provision_!) )
    ( Satellite (leaf 90) (rel2par elaboration-object-attribute-e) (text _!concerning the release of toxic
      substances into the environment._!) )
)
( Nucleus (span 91 92) (rel2par span)
    ( Nucleus (leaf 91) (rel2par span) (text _!In 1986 Congress strengthened the penalty_!) )
    ( Satellite (leaf 92) (rel2par means) (text _!by making it a felony.<P>_!) )
)
```

Figure 8. BACKGROUND/background example from wsj_1331 (Carlson et al. 2002).

"Circumstance" relations are syntactically and structurally similar to many of the EVALUATION and EXPLANATION relations when there are no cue words present to signal the type of relation, contributing to the confusion.

Cause

None of the CAUSE samples were identified using *Label*, and 66.67% of them were misclassified as ELABORATION. Once ELABORATION is removed, identification shows some improvement.

Label_NoElab frequently misclassified CAUSE relations as EVALUATION, EXPLANATION, and BACKGROUND, prompting further investigation into the CAUSE relation group. Some of the relations included in the CAUSE class have definitions that overlap with relations in other classes. For instance, to the relation "reason" which falls under the EXPLANATION class is nearly identical to the CAUSE relations.

Comparison

COMPARISON had poor performance using *Label* because it was misclassified as ELABORATION. Removing ELABORATION with *Label_NoElab*, however, only revealed that COMPARISON instances were being classified as anything and everything. The features that should assist the classifier in identifying COMPARISON relations, namely the Penn Treebank tags JJS, RBS, JJR, and RBR (see Section 3.3.1), have a stronger presence in other classes.

In all of the trials of this study, COMPARISON has consistently done poorly. This seems odd since COMPARISON relations are some of the easiest for humans to detect. Lin et al. (2009) acknowledge COMPARISON as one of the more difficult discourse relations to machine identify. Hernault et al. (2010) report an F-score of 10.5% on COMPARISON relations that have a nuclearity of N-S. The other relations

in the COMPARISON class were not included in their results, although they are present in the test set of the corpus.

Evaluation and Explanation

The accuracy of EVALUATION only suffers when ELABORATION is present, likely because the two are so similar. *Label* placed 63 of the 71 EVALUATION samples under ELABORATION. While performance on the EVALUATION class improved tremendously when using *Label_NoElab*, precision was only 30.7%. Low precision indicates that the class had a large number of false positives. Most of the false positives were misclassified EXPLANATION instances.

EXPLANATION performance improved slightly with *Label_NoElab*, but it too yielded a low precision (32.7%). This class seems to serve as the "other" class in the absence of ELABORATION.

EVALUATION and EXPLANATION relations are difficult to distinguish from one another, even for humans. EVALUATION contains the relation "interpretation" which is defined in the Discourse Tagging Reference Manual as, "… an explanation of what is not immediately plain or explicit," (Carlson et al. 2001a). The "conclusion" relation under EVALUATION is similar to the "explanation-argumentative" relation under EXPLANATION. From a syntactic standpoint, the two are classes are identical. The difference is implicit and semantic. EXPLANATION relations are more objective and factual; EVALUATION relations are more subjective and may reflect the opinion of the author or speaker.

Summary

SUMMARY experienced poor performance using both *Label* and *Label_NoElab*. This is appears to be due to its small sample size and its overlap with other relation classes like EVALUATION, ELABORATION, and BACKGROUND.

Temporal

TEMPORAL relations can be split into two groups: temporal ("temporal-before", "temporal-same-time", and "temporal-after") and sequence ("sequence" and "inverted-sequence"). The temporal relations are often signaled by cues such as "before", "while", or "after". These relations were nearly all misclassified as BACKGROUND. This was not surprising since the relations in the BACKGROUND class are signaled by similar words.

The multinuclear sequence relations are events listed in chronological or reverse chronological order and were all misclassified as JOINT. Much of this could have been caused by the method for extracting the cue word features. Many of the sequence relations have a structure similar to the example show in Figure 9. The cue word "and", a common signal for JOINT relations, at the beginning of the second span prevents the more helpful cue word "then" from being detected.

```
( Nucleus (leaf 141) (rel2par Sequence) (text _!"They started,_!) )
( Nucleus (leaf 142) (rel2par Sequence) (text _!and then abandoned it."_!) )
```

Figure 9. TEMPORAL/Sequence example from wsj_1146 (Carlson et al. 2002).

The output from the k-NN classifier *Label* showed that for a majority of the TEMPORAL instances, the TEMPORAL class was the second-nearest neighbor, meaning the samples would have been classified correctly in the absence of BACKGROUND or JOINT.

Topic-Comment

Label was unable to classify any TOPIC-COMMENT samples. In each instance, even if the classifier had not selected the incorrect label, TOPIC-COMMENT still would not have been the next choice. TOPIC-COMMENT was actually one of the *farthest* neighbors. The TOPIC-COMMENT class had the smallest sample size in the training data (148 instances). With the variety of relations contained in the group, a sample size of 148 is simply not enough training data for a classifier to learn it.

Chapter 5 – Conclusions and Future Research

To date, the only open source discourse parser is SPADE, which is limited to identifying discourse relations within a single sentence (Soricut et al. 2003). HILDA is a discourse parser not publicly available which allegedly performs automatic discourse structure identification at the document level (Hernault et al. 2010), but at the cost of accuracy. This study expands automatic discourse identification to the document level, without a loss in performance. A summary and comparison of the results (calculated F-scores) from all three studies are provided in Table 14. Also included in the Table 14 are F-scores for human agreement for each of the tasks, as calculated by Hernault et al. (2010).

Table 14. Summary of performance compared to SPADE, HILDA, and humans

	This study	SPADE	HILDA	Human agreement
Structure	94.8%	92.8%	83.0%	88.1%
Nuclearity	70.8%	N/A	68.4%	77.5%
Labeling	64.5%	63.8%	55.3%	66.0%

Before any further research is conducted in this area, the grouping of relation labels needs to be evaluated carefully and modified. There are many relations, such as “reason” and “comment”, which could belong to multiple groups, contributing to confusion and poor performance. The RST classes should avoid having multiple miscellaneous classes like ELABORATION. If possible, classes should not contain sub-relations that are too structurally or semantically dissimilar. An example of this dissimilarity is found in the TOPIC-COMMENT relations “question-answer” and “comment-topic”.

The next step is to investigate additional features that will better identify the relation classes with low performance. Incorporating span polarity and affect recognition as features may assist in distinguishing between the objective and subjective classes like EXPLANATION and EVALUATION. It may also improve performance of other classes by signaling changes in the author’s tone across spans.

More focus should be put on boundary words, expanding the word group features (Figure 4) based on obvious patterns across relation classes. The word groups should prove better than distinct words (Table 9) in identifying both nuclearity and relation labels. Using distinct words as features prevents the trained classifier from clustering data as humans do naturally.

The method for extracting cue words needs to be improved. This study limited each span to one cue word, selecting the cue nearest the relation boundary when more than one was present. On the left boundary of the right span, there are often less helpful cue words (“and”, “for”) preceding the stronger cue words which serve as better signals for specific relation classes (“even”, “before”). Cue word features should be extracted differently to prevent the loss of internal, more useful cues.

Finally, this study has found limitations with the use of the RST Discourse Treebank corpus for discourse structure identification. It is a corpus of periodicals, full of abbreviations, shorthand, and strange formatting. The subject range is narrow, only addressing business and financial news. The articles contain more numeric values and business terminology that what is used in natural language. Because of these limitations, models trained on this corpus risk are at a risk over-fitting. Additionally, with the limited number of samples for some relation label classes, there is no way to train a classifier to predict

them with high accuracy. Unfortunately, the RST Discourse Treebank is the largest and most widely accepted corpus available at this time for discourse.

References

Attali, Y. and J. Burstein (2006). Automated essay scoring with e-rater v.2. *Journal of Technology, Learning, and Assessment* **4**(3).

Carlson, L. and D. Marcu (2001a). Discourse Tagging Reference Manual. *ISI Tech Report ISI-TR-545***:** 87.

Carlson, L., D. Marcu, et al. (2001b of Conference). Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. Proceedings from *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*. Aalborg, Denmark, Association for Computational Linguistics.

Carlson, L., D. Marcu, et al. (2002). RST Discourse Treebank. Linguistic Data Consortium. Philadelphia.

Chang, C.-C. and C.-J. Lin (2011). LIBSVM : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2**(3): 1-27.

Cristea, D., N. Ide, et al. (1999). Discourse structure and co-reference: an emperical study. Proceedings from *The 18th International Conference on Computational Linguistics COLING'2000*. Luxembourg.

ETS. (2011). "Automated scoring and natural language processing. Retrieved from http://ets.org/research/topics/as_nlp."

Finkel, J. R., T. Grenager, et al. (2005). Incorporating non-local information into information extraction systems by Gibbs Sampling. Proceedings from *The 43rd Annual Meeting of the Association for Computational Linguistics ACL 2005*. Ann Arbor, USA.

Hernault, H., H. Prendinger, et al. (2010). HILDA: a discourse parser using support vector machine classification. *Dialogue & Discourse* **1**(3): 1-33.

Hovy, E. H. (1994). Automated discourse generation using discourse structure relations. *Natural Language Processing*, MIT Press**:** 341-385.

Jurafsky, D. and J. H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, Pearson Education, Inc.

Klein, D. and C. D. Manning (2003). Accurate unlexicalized parsing. Proceedings from *The 41st Annual Meeting of the Association for Computational Linguistics ACL 2003*. Sapporo, Japan.

Lin, Z., M.-Y. Kan, et al. (2009 of Conference). Recognizing implicit discourse relations in the Penn Discourse Treebank. Proceedings from *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*. Singapore, Association for Computational Linguistics.

Lin, Z., H. T. Ng, et al. (2011). Automatically evaluating text coherence using discourse relations. Proceedings from *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, Association for Computational Linguistics.

Louis, A., A. Joshi, et al. (2010 of Conference). Discourse indicators for content selection in summarization. Proceedings from *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Tokyo, Japan, Association for Computational Linguistics.

Mann, W. C. and S. A. Thompson (1988). Rhetorical structure theory: toward a functional theory of text organization. *Text* **8**(3): 243-281.

Marcu, D. (1997). From discourse structures to text summaries. Proceedings from *The ACL Workshop on Intelligent Scalable Text Summarisation*. Madrid, Spain.

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*, The MIT Press.

Marcu, D. and A. Echihabi (2002 of Conference). An unsupervised approach to recognizing discourse relations. Proceedings from *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Philadelphia, Pennsylvania, Association for Computational Linguistics.

Marir, F. and K. Houam (2002 of Conference). RSTIndex: indexing and retrieving web document using computational and linguistic techniques. Proceedings from *The Third International Conference on Intelligent Data Engineering and Automated Learning*, Springer-Verlag.

Mochales, R. and M.-F. Moens (2011). Argumentation mining. *Artificial Intelligence and Law* **19**(1): 1-22.

Penn Discourse Treebank Research Group (2007). The penn discourse treebank 2.0 annotation manual.

Pitler, E., A. Louis, et al. (2009 of Conference). Automatic sense prediction for implicit discourse relations in text. Proceedings from *The Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*. Suntec, Singapore, Association for Computational Linguistics.

Reitter, D. (2003). *Rhetorical analysis with rich-feature support vector models*. PhD, University of Potsdam.

Rosner, D. and M. Stede (1992). Customizing rst for the automatic production of technical manuals. *Aspects of automated natural language generation*(FAW- TR-91028): 199-214.

Sagae, K. (2009 of Conference). Analysis of discourse structure with syntactic dependencies and data-driven shift-reduce parsing. Proceedings from *The 11th International Conference on Parsing Technologies*. Paris, France, Association for Computational Linguistics.

Soricut, R. and D. Marcu (2003 of Conference). Sentence level discourse parsing using syntactic and lexical information. Proceedings from *The 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*. Edmonton, Canada, Association for Computational Linguistics.

Subba, R. and B. Di Eugenio (2007). Automatic discourse segmentation using neural networks. Proceedings from *The 11th Workshop on the Semantics and Pragmatics of Dialogue*. Rovereto, Italy.

Subba, R. and B. Di Eugenio (2009 of Conference). An effective discourse parser that uses rich linguistic information. Proceedings from *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Boulder, Colorado, Association for Computational Linguistics.

Taboada, M. and J. Grieve (2004). Analyzing appraisal automatically. Proceedings from *The AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. Stanford, USA.

Thanh, H., G. Abeysinghe, et al. (2004). Automated discourse segmentation by syntactic information and cue phrases. *Artificial Intelligence and Applications*.

Tofiloski, M., J. Brooke, et al. (2009 of Conference). A syntactic and lexical-based discourse segmenter. Proceedings from *The ACL-IJCNLP 2009 Conference Short Papers*. Suntec, Singapore, Association for Computational Linguistics.

Vita

Jamie Allison Guidry was born in Breaux Bridge, Louisiana, in 1986. Following her graduation from Dutchtown High School, Geismar, Louisiana, she pursued and obtained a bachelor's of science in industrial engineering from Louisiana State University in 2010. She began working toward the degree of master's of science in engineering science with concentration in information technology and engineering at Louisiana State University in 2011. During her time as a graduate student, she worked as a graduate assistant for the Louisiana State University E. J. Ourso College of Business Office of Alumni and External Relations. Her degree will be conferred at the Fall Commencement December 2012.