Neural Discourse Modeling

A Dissertation

Presented to

The Faculty of the Graduate School of Arts and Sciences

Brandeis University

Computer Science

Nianwen Xue, Advisor

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

by

Attapol Thamrongrattanarit Rutherford

August, 2016

This dissertation, directed and approved by Attapol Thamrongrattanarit Rutherford's committee, has been accepted and approved by the Graduate Faculty of Brandeis University in partial fulfillment of the requirements for the degree of:

DOCTOR OF PHILOSOPHY

Eric Chasalow, Dean
Graduate School of Arts and Sciences

Dissertation Committee:
Nianwen Xue, Chair
James Pustejovsky, Dept. of Computer Science
Benjamin Wellner, Dept. of Computer Science
Vera Demberg, Universität des Saarlandes

Acknowledgments

This dissertation is the most difficult thing I have ever done in my life. As lonely as a dissertation might seem, I manage to get to this point thanks to the help and support from many people.

Bert obviously for all your advice and support. And also for putting up with me all these years. I know I am not the easiest student to deal with in many ways, but you calm me down and help me grow up as an adult as a researcher. I cannot thank you enough for that.

John for keeping me sane and really being there for me. Time and time again you rescue me from hitting the PhD emotional rock bottom. Thanks for being the awesome person that you are. I would have been a wreck without you.

Vera for a new perspective on the the work and the world. My time at Saarland completely changes the way I think about the discourse analysis and computational linguistics in general. The opportunity to meet with the brilliant colleagues in Germany and the opportunity to explore Europe from the Saarbrücken base camp are simply priceless.

James and Ben for being on the committee and touching base with me throughout the process. Yaqin, Chuan, and Yuchen for being supportive lab mates. I appreciate both moral and intellectual support through these years.

Abstract

Neural Discourse Modeling

A dissertation presented to the Faculty of
the Graduate School of Arts and Sciences of
Brandeis University, Waltham, Massachusetts

by Attapol Thamrongrattanarit Rutherford

Sentences rarely stand in isolation, and the meaning of the text is always more than the sum of its individual sentences. Sentences form a discourse and require the analysis beyond the sentence level. In the analysis of discourse, we must identify discourse relations that hold between a pair of textual spans. This is a simple task when discourse connectives such *because* or *therefore* are there to provide cues, but the relation must be inferred from the text alone when discourse connectives are omitted, which happens very frequently in English. Discourse parser may be used as another preprocessing step that enhances downstream tasks such as automatic essay grading, machine translation, information extraction, and natural language understanding in general.

In this thesis, we propose a neural discourse parser which addresses computational and linguistic issues that arise from implicit discourse relation classification in the Penn Discourse Treebank. We must build a model that captures the inter-argument interaction without making the feature space too sparse. We shift from the traditional discrete feature paradigm to the distributed continuous feature paradigm, also known as neural network modeling. Discrete features used in previous work reach their performance ceiling because of the sparsity problem. They also cannot be extended to other domains or languages due their dependency on specific language resources and lexicons. Our approaches are language-independent and generalizable features, which reduce the feature set size by one or two orders of magnitude. Furthermore, we deal with the data sparsity problem directly through

linguistically-motivated selection criteria for eligible artificial implicit discourse relations to supplement the manually annotated training set. Learning from these insights, we design the neural network architecture from the ground up to understand each subcomponent in discourse modeling and to find the model that performs the best in the task. We found that high-dimensional word vectors trained on large corpora and compact inter-argument modeling are integral for neural discourse models. Our results suggest simple feedforward model with the summation of word vectors as features can be more effective and robust than the recurrent neural network variant. Lastly, we extend these methods to different label sets and to Chinese discourse parsing to show that our discourse parser is robust and language-independent.

Contents

Abstract **v**

1 Introduction **3**
1.1 Motivation 5
1.2 Dissertation Contributions 6
1.3 Robust Light-weight Multilingual Neural Discourse Parser 10

2 Discourse Parsing and Neural Network Modeling in NLP **11**
2.1 Discourse Analysis 11
2.2 Modeling Implicit Discourse Relations in the PDTB 18
2.3 Neural Network Modeling 20

3 Brown Cluster Pair and Coreference Features **31**
3.1 Sense annotation in Penn Discourse Treebank 33
3.2 Experiment setup 34
3.3 Feature Description 35
3.4 Results 39
3.5 Feature analysis 40
3.6 Related work 45
3.7 Conclusions 46

4 Distant Supervision from Discourse Connectives **50**
4.1 Discourse Connective Classification and Discourse Relation Extraction . . . 53
4.2 Experiments 60
4.3 Related work 68
4.4 Conclusion and Future Directions 70

5 Neural Discourse Mixture Model **71**
5.1 Corpus and Task Formulation 73
5.2 The Neural Discourse Mixture Model Architecture 74
5.3 Experimental Setup 80

5.4 Results and Discussion 83
5.5 Skip-gram Discourse Vector Model 85
5.6 Related work 88
5.7 Conclusion 89

6 Recurrent Neural Network for Discourse Analysis 91
6.1 Related Work 93
6.2 Model Architectures 94
6.3 Corpora and Implementation 98
6.4 Experiment on the Second-level Sense in the PDTB 100
6.5 Extending the results across label sets and languages 104
6.6 Conclusions and future work 107

7 Robust Light-weight Multilingual Neural Parser 108
7.1 Introduction 108
7.2 Model description 110
7.3 Experiments 111
7.4 Error Analysis 113
7.5 Conclusions 115

8 Conclusions and Future Work 116
8.1 Distant Supervision with Neural Network 117
8.2 Neural Network Model as a Cognitive Model 118
8.3 Discourse Analysis in Other Languages and Domains 118
8.4 Concluding Remarks 119

List of Figures

2.1 Example of a structure under Rhetorical Structure Theory 12
2.2 Example of discourse relations in the Penn Discourse Treebank. 16
2.3 System pipeline for the discourse parser 17
2.4 Fully connected feedforward neural network architecture with one hidden layer. 20
2.5 Skip-gram word embedding 23
2.6 Recurrent neural network architecture 26
2.7 Discourse-driven RNN language model 30

3.1 Bipartite graphs showing important features 48
3.2 Coreferential rates in the PDTB 49

4.1 Omission rates across discourse connective types 61
4.2 Distributions of Jensen-Shannon context divergence 63
4.3 Scattergram of all discourse connectives 64
4.4 The efficacy of different discourse connectives classes in distant supervision . 65
4.5 Comparing distant supervision from the PDTB and Gigaword as sources . . 68

5.1 Neural Discourse Mixture Model 74
5.2 Skip-gram architecture 75
5.3 Mixture of Experts model 78
5.4 Histogram of KL divergence values 87

6.1 Feedforward and LSTM discourse parsers 94
6.2 Summation pooling gives the best performance in general. The results are shown for the systems using 100-dimensional word vectors and one hidden layer. 101
6.3 Inter-argument interaction 101
6.4 Comparison between feedforward and LSTM 102
6.5 Comparing the accuracies across Chinese word vectors for feedforward model. 107

7.1 Light-weight robust model architecture 110

List of Tables

2.1 The relation definitions in Rhetorical Structure Theory 13
2.2 Sense hierarchy in the Penn Discourse Treebank 2.0 15
2.3 Discourse parser performance summary 17

3.1 The distribution of senses of implicit discourse relations is imbalanced. . . . 33
3.2 Brown cluster pair feature performance 37
3.3 Ablation studies of various feature sets 38

4.1 Discourse connective classification 62
4.2 The distribution of senses of implicit discourse relations in the PDTB 62
4.3 Performance summary of distant supervision from discourse connectives in 4-way classification 66
4.4 Performance summary of distant supervision from discourse connectives in binary classification formulation 67
4.5 Comparison of different discourse connective classes 67
4.6 The sense distribution by connective class. 69

5.1 Tuned hyperparameters of the models 83
5.2 Performance summary of the Neural Discourse Mixture Model 84
5.3 Clustering analysis of Skip-gram discourse vector model 90

6.1 The distribution of the level 2 sense labels in the Penn Discourse Treebank . 98
6.2 Performance comparison across different models for second-level senses. . . . 100
6.3 Results for all experimental configuration for 11-way classification 102
6.4 Chinese model summary 106

7.1 Sense-wise F_1 scores for English non-explicit relations 113
7.2 Sense-wise F_1 scores for Chinese non-explicit discourse relation. 114
7.3 Confusion pairs made by the model 114

Chapter 1

Introduction

Discourse analysis is generally thought of as the analysis of language beyond the sentence level. Discourse-level phenomena affect the interpretation of the text in many ways as sentences rarely stand in isolation. When sentences form a discourse, the semantic interpretation of the discourse is more than the sum of individual sentences. The ordering of sentences within affects the interpretation of the discourse. Alteration of sentence ordering may render the discourse incoherent or result in a totally different interpretation. The discourse-level phenomena and text coherence have repercussions in many Natural Language Processing tasks as the research community shifts its focus away from the sentence-level analysis. Automatic discourse analysis is crucial for natural language understanding and has been shown to useful for machine translation, information extraction, automatic essay grading, and readability assessment. A discourse parser that delineates how the sentences are connected in a piece of text will need to become a standard step in automatic linguistic analysis just like part-of-speech tagging and syntactic parsing.

Sentences must be connected together in a specific way to form a coherent discourse within the text. To compose a paragraph that "makes sense," the sentences must form

a discourse relation, which manifests the coherence between the sentences. The discourse relation that glues together a pair of sentences can be explicitly signaled by a discourse connective or implicitly inferred from the text as long as the sentences are semantically coherent. Examples of discourse relation are demonstrated below:

(1) I want to add one more truck. (Because) I sense that the business will continue to grow.

(2) # I do not want to add one more truck. (Because) I sense that the business will continue to grow.

The discourse relation (1) does not require discourse connectives *because* to signal a causal relation. The causal relation can be inferred from the text, and the discourse sounds natural regardless of the use of *because*. However, the pair of sentences in (2) do not form a coherent discourse relation, and the sense of discourse relation cannot be inferred with or without the discourse connective.

In this dissertation, we focus on the implicit discourse relation classification where the discourse relations are not signaled explicitly by discourse connectives. Our main goal is to develop novel approaches that determine the sense of discourse relation that holds between a pair of textual spans. We devise various algorithms from traditional feature engineering paradigm, distant-supervision learning paradigm, and neural network modeling. These algorithms address linguistic and computational challenges associated with the task of implicit discourse relation classification. This chapter will discuss the main motivation of our work and provide the overview of our contributions so far in addressing these issues.

1.1 Motivation

Our work focuses on the implicit discourse relation classification because it is the performance bottleneck of a discourse parser. The performance of a typical discourse parser on the explicit discourse relations hovers around 90% accuracy, but the performance on the implicit discourse relations reaches measly 40% accuracy (Xue et al., 2015). Arguably, implicit discourse relation classification is the last component missing from a fully function discourse parser, which can then be used in many natural language processing application. In other words, we are only one step away from having a discourse parser.

The sense of an implicit discourse relation is much more difficult than an explicit one due to the lack of discourse connectives, which provide high fidelity signal of the senses. By mapping each discourse connective to its most likely sense, we can achieve as high as 82% accuracy on the explicit discourse relations. The situations are much more complicated for implicit discourse relations as we can no longer rely on such signal. We must infer the sense solely from the text, its context, and other external linguistic resources. The complexity of implicit discourse relation parser is much higher than the explicit counterpart. Statistical approaches to this problem have seen limited success because the task is plagued by the sparsity problem, complicated by wild linguistic variability, and limited by the size of the dataset (Pitler et al., 2009). This task requires discourse relation representation that can capture the patterns and regularities of discourse relations despite data scarcity. The system must utilize compact representation while using sophisticated model formulation.

We aim to remedy this problem by reducing the sparsity of the feature space directly. A traditional machine learning approach to such NLP task employs around a million of features. A lot of the features contribute more noise than signal to the system (Pitler et al., 2009; Lin et al., 2009) because the variability is high, but the size of the dataset is

small. Feature selection becomes a crucial step in improving the system (Park and Cardie, 2012). This problem motivates us to find a novel alternative to sparse feature space and also artificially increase the size of the training set without extra annotated corpus.

We also explore distributed representation or neural network modeling and move away from the manual feature engineering, which causes the problem at the first place. Neural network models have revolutionized speech recognition and some branches of NLP within the past recent years. This alternative representation does not rely so heavily on feature engineering and feature selection. It instead relies on non-linear transformation and extracting semantic information from unannotated data. Under this paradigm, we do not have to handcraft our own features or conduct complicated feature selection.

1.2 Dissertation Contributions

This dissertation is concerned with assigning a pair of sentences with the discourse relation sense that hold within the pair in the style of the Penn Discourse Treebank. We attack this problem from different angles inspired by the weaknesses of previous approaches and the linguistic issues in discourse modeling. The major contributions of this dissertation are summarized as follows.

1.2.1 Generalizable Featurization

In this work, we create more generalizable featurization to alleviate sparsity problem in discourse modeling. The detail of this work can be found in Chapter 3. We use Brown clustering algorithm to create compact generalizable features and employ coreferential patterns that exist through the discourse. Since the publication of this work, Brown cluster pair features have once been part of the state-of-the-art system, and has now been considered a standard

strong baseline feature set by a few other subsequent work in discourse parsing (Xue et al., 2015; Ji and Eisenstein, 2015; Braud and Denis, 2015; Yoshida et al., 2015; Wang et al., 2015). This feature set has many advantages. It is very simple to generate and reproduce as the unannotated corpus is the only ingredient. It also does not depend on specific linguistic resources, so this feature set can also be applied to other languages as well.

The algorithm works as follows. We first cluster all of the words based on the unannotated data and use the cluster assignment as the basis for feature engineering. The features that are known to be effective in this task derive from an interaction of features across the pair of sentences. For example, cartesian products of the words in the pair, the conjunction of the verb classes of the main verbs. These features often create too many features that contain little signal, and they require external resources such as verb class lexicon. Brown clustering is our algorithm of choice as it has been shown to work well in various NLP tasks while maintaining reasonable complexity. We can then employ cartesian product of Brown clusters as the main feature set. Our experiments suggest that this feature set is effective in combination with other surface features and improves the state-of-the-art at the time. Another related feature set that we use takes advantage of in this work is the coreference chain within the discourse. We generate a feature set that takes into account the entities that exist within and across the sentences. When paired with Brown cluster representation, coreferential pattern features contribute to a marginal improvement in performance.

1.2.2 Distant Supervision from Discourse Connectives

In this work, we design and test the selection criteria for eligible extra artificial training data through the computational study of discourse connectives in the PDTB to further alleviate the sparsity problem. This is the first successful effort in incorporating explicit discourse

relations and unannotated data in enhancing the performance of implicit discourse relations, to the best of our knowledge.

Discourse connectives provide high-fidelity signal for the sense of discourse relations hence making the text easy to understand for both human and the machine. Unfortunately, implicit discourse relations do not benefit from such signal because the discourse connectives are omitted. It is natural to think that we can simply artificially omit all discourse connectives and use the explicit discourse relations as extra training data for implicit discourse relation classifier. However, implicit discourse relations are not equivalent to such explicit discourse relations as confirmed by the dramatic drop in performance (Pitler et al., 2009).

Besides the use of discourse connectives, in what way do explicit and implicit discourse discourse relations differ? We hypothesize that certain discourse connectives are disproportionately omitted. In other words, only certain explicit discourse relations are equivalent to the implicit discourse relations when the discourse connectives are omitted. If we have a way to identify such relations, we can artificially increase the size of the dataset and increase our capability to train a larger model. We study the linguistic and statistical properties of the English discourse connectives and classify them into classes. We discover *freely omissible discourse connectives* and use them to select instances of explicit discourse relations from both annotated and unannotated data to include in the training set. Only this class of discourse connective improves the performance. The addition of these artificial data effectively reduces the data scarcity and feature sparsity problems.

1.2.3 Dense Featurization and Neural Network Modeling

We break away from sparse featurization and use distributed features that do not suffer from the sparsity problem in order to improve the accuracy. We explore the use of distributional

semantics and neural network modeling the domain of discourse relation classification. We take advantage of word embedding induced on a large unannotated corpus and essentially construct dense feature vectors to represent discourse relations. We name our first model Neural Discourse Mixture Model (NDMM), which combines the strength of surface features and continuous features to capture the variability and create abstract features that we cannot otherwise discover. The NDMM further improves the classification performance from our previous work.

One of the surprising findings from this work is that distributed continuous features constructed from word embedding alone can achieve the performance once only afforded by handcrafted features. This is made possible the use of non-linear transformation implemented by the network. Upon further computational analysis of feature vectors, we found that the network indeed abstract features such as topic similarity, topic shift, and sentiment gradient, which provides some explanation for why this approach can quite effective in discourse analysis.

1.2.4 Sequential and Tree-structured Recurrent Neural Network for Discourse Parsing

We are the first to build Long Short-Term Memory (LSTM) Model constructed from the sequential and the tree structures of the sentences in discourse relations. Through experimentation with these models, we investigate the effects and the interaction between lexical knowledge and structural knowledge encoded in the discourse relation. Our best neural model achieves the performance equivalent to the previous best-performing surface features.

LSTM models are superior to the feedforward neural network in that it can capture the sequential information infinitely back in time (Hochreiter and Schmidhuber, 1997; Gers et

al., 2000). They also learn what to remember and what to forget in the previous time steps (Sundermeyer et al., 2012a). Moreover, it can be structured from a parse tree to encode syntactic information (Tai et al., 2015). We hypothesize that as we employ more rigid structures to the construction of feature vectors, the performance will improve. The small set of parameters and more structured model should allow us to be more data efficient.

1.3 Robust Light-weight Multilingual Neural Discourse Parser

Lastly, we extend what we learn from discourse modeling in the PDTB onto the corpus of Chinese Discourse Treebank (CDTB) and test how robust our approach is when the discourse parser must handle different label sets. We show that feedforward neural architecture with the summation of individual word vectors as features is robust to different label sets and different languages. This model is easy to deploy and re-train as it does not need to generate large feature vectors or carry around many parameters. The performance of our approach on the CDTB and various adaptation of PDTB is comparable if not better than a system loaded with surface features although our approach does not require external resources such as semantic lexicon or a syntactic parser.

Chapter 2

Discourse Parsing and Neural Network Modeling in NLP

This chapter provides the overview of two dominant theories in computational discourse analysis with emphasis on the analysis in the style of the Penn Discourse Treebank. We will also survey previous approaches in discourse parsing and recent successes of neural network modeling in discourse parsing and natural language processing in general.

2.1 Discourse Analysis

If we think of syntax as a study of how words are pieced together to form a grammatical sentence, we can think of discourse analysis as a study of how sentences are pieced together to form a coherent body of text. A lot of work has been done to study discourse in the context of a dialogue or a coherent turn-taking conversation. In this dissertation, we focus on written text or monologue, where there is not explicit turn-taking between agents in the discourse.

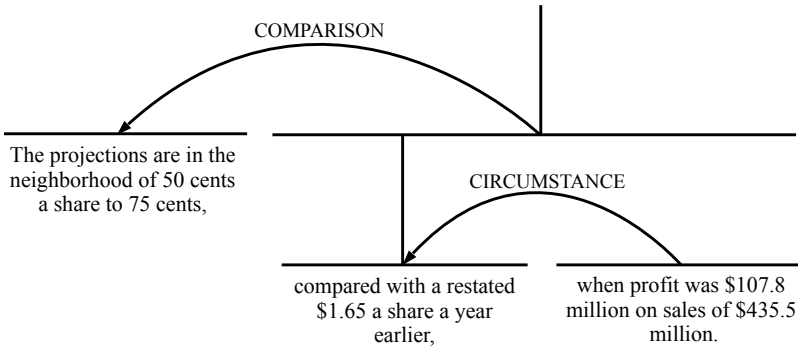

Figure 2.1: Example of a partial structure under Rhetorical Structure Theory. The text spans are the *elementary discourse units*. An arrow originates at the *nucleus* and points to the *satellite*. (Taken from Ji & Eisenstein (2014))

However, unlike syntax, there do not exist clean theories of discourse grammars that explain or describe how a sequence of sentences make a coherent text. We have not seen discourse features or subcategories that can fully characterize and explain discourse phenomena the same way we do in syntax. There are a few theories along with annotated corpora that illustrate the theories and allow for computational investigation of discourse. We will discuss two theories and their corresponding corpora namely Rhetorical Structure Theory and the Penn Discourse Treebank.

Rhetorical Structure Theory

Rhetorical Structure Theory (RST) adopts a tree as the structure that underlies relationships within units of text (Mann and Thompson, 1988). As the name suggests, the discourse analysis of this style imposes the structure of the text where all of the textual units are completely interconnected. In terms of structure, the smallest units of text in this theory is called *elementary discourse unit* (EDU). An EDU does not correspond to any syntactic constituents as a syntactic construct might not map nicely to a discourse construct and there

Circumstance	Antithesis and Concession
Solutionhood	Antithesis
Elaboration	Concession
Background	Condition and Otherwise
Enablement and Motivation	Condition
Enablement	Otherwise
Motivation	Interpretation and Evaluation
Evidence and Justify	Interpretation
Evidence	Evaluation
Justify	Restatement and Summary
Relations of Cause	Restatement
Volitional Cause	Summary
Non-volitional Cause	Other Relations
Volitional Result	Sequence
Non-volitional Result	Contrast
Purpose	

Table 2.1: The relation definitions in Rhetorical Structure Theory first proposed by Mann & Thompson (1988)

is no compelling reason to believe that discourse analysis should comply to one particular syntactic theory. The arguments of the relation are a pair of EDUs functionally named as nucleus and satellite. Some of the relations also allow for multiple satellites. Under these schema, the set of relations form a tree that underlie the structure of the text (Figure 2.1). In terms of functional properties and organization of discourse in RST, the original proposal defines 24 primary types of rhetorical relations (Table 2.1). However, this definition of rhetorical senses is designed to be an open set, where one can extend or modify for particular kind of analysis or application.

The strengths of the RST for discourse modeling are the structure imposed by the analysis and the annotated corpora that are large enough for computational modeling (Carlson et al., 2003). The complete tree structure describes precisely how one EDU relates to another in the text regardless of the textual distance. As restrictive as this analysis seems, majority of

texts can be analyzed in this fashion with the exception of laws, contracts, reports, and some forms of creative writing. This type of description is validated through a large annotated corpus, which opens door for computational studies of discourse and automated discourse analysis. The RST discourse parser has shown promises for downstream NLP applications such as text generation and summarization (Marcu, 1999). The research community has been actively working on improve the performance of automatic discourse parser in the style of RST (Ji and Eisenstein, 2014; Feng and Hirst, 2014; Feng and Hirst, 2012).

The Penn Discourse Treebank

The Penn Discourse Treebank models a discourse as a list of binary discourse relations in contrast to the tree structure imposed by the RST (Prasad et al., 2008; Prasad et al., 2007). The PDTB does not have the smallest textual units where the analysis can be built in a bottom fashion. On the contrary, the textual argument can be any congruent or non-congruent text spans and is called *abstract object*, which describes states, events, or propositions. A discourse relation only shows the local discourse coherence as not all text spans are connected in this kind of analysis. The structure of PDTB analysis is less rigid than the one of the RST.

Each discourse relation consists of the first argument *Arg1*, the second argument *Arg2*, a discourse connective, and a sense tag. The discourse analysis in the style of the PDTB is said to be *lexically grounded* because the annotation must be based on the discourse connectives whether or not they are omitted in the text. Arg2 is the textual argument that is syntactically bound to the discourse connective. If the discourse connective is present, it will be annotated as such. If not, the most plausible connective are artificially inserted by the annotators. The sense annotation is also lexically grounded in that it must be determined solely based on

TEMPORAL	COMPARISON
Asynchronous	Contrast
Motivation	juxtaposition
precedence	opposition
succession	*Pragmatic Contrast*
	Concession
	expectation
CONTINGENCY	contra expectation
Cause	Pragmatic Concession
reason	
result	EXPANSION
Pragmatic Cause	Conjunction
justification	Instantiation
Condition	Restatement
hypothetical	specification
general	equivalence
unreal present	generalization
unreal past	Alternative
factual present	conjunctive
factual past	disjunctive
Pragmatic Condition	chosen alternative
relevance	Exception
implicit assertion	List

Table 2.2: Sense hierarchy in the Penn Discourse Treebank 2.0

the sense of discourse connectives disambiguated by the arguments as necessary. Figure 2.2 shows an example of the discourse relations. Note that the relations do not form a tree structure although the arguments of the two discourse relations might overlap and that the discourse connectives must be annotated for all relations regardless of whether they are actually in the text or not to enforce lexical groundedness.

The contrast between RST and PDTB with respect to relation types has an implication on the modeling choice. The sense inventory in RST is mostly flat and designed to have high coverage while allowing more senses to be added to the set as we move to different domains or specific applications. On other hand, the PDTB sense inventory is organized

Explicit discourse relation (Comparison.Concession)
According to Lawrence Eckenfelder, a securities industry analyst at Prudential-Bache Securities Inc., "Kemper is the first firm to make a major statement with program trading." He added that "**having just one firm do this isn't going to mean a hill of beans**. But *if this prompts others to consider the same thing, then it may become much more important.*"

Implicit discourse relation (Comparison.Contrast)
According to Lawrence Eckenfelder, a securities industry analyst at Prudential-Bache Securities Inc., "**Kemper is the first firm to make a major statement with program trading**." He added that "[however] *having just one firm do this isn't going to mean a hill of beans.* But if this prompts others to consider the same thing, then it may become much more important."

Figure 2.2: Example of discourse relations in the Penn Discourse Treebank. The bold faced text span is Arg1, the italicized text span is Arg2, and the underlined text span is discourse connective. The connective in the square brackets are inserted by the annotators.

in a hierarchical fashion (Figure 2.2). The sense hierarchy is beneficial in that it allows the annotation to be more reliable as the annotators can move up and down the hierarchy as much as the information in the text allows them to. At the same time, it allows for finer-grained annotation, which can be essential for certain analysis or application.

In addition to its lexical groundedness and hierarchy of senses, the PDTB is ideal for computational modeling because the annotated corpus is the largest collection of discourse annotation. The source of the corpus is from the Wall Street Journal, the same source text as the Penn Treebank (Marcus et al., 1993). In other words, the PDTB is another annotation layer over the Penn Treebank. We can use the gold standard parse trees when testing the systems that utilize the features derived from syntactic trees to gauge how much the performance is affected by the mistakes made by the automatic syntactic parser. As discourse phenomena have more variation and the theoretical underpinning of discourse is not as solidified as syntactic parsing, discourse parsing requires a large amount of data to reach an acceptable performance (Pitler et al., 2008; Pitler et al., 2009).

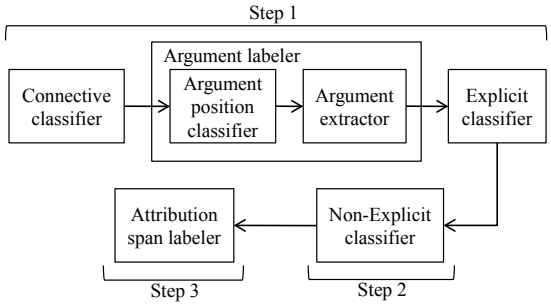

Figure 2.3: System pipeline for the discourse parser (Lin et al., 2014)

	Explicit parser			Implicit parser		
	Precision	Recall	F_1	Precision	Recall	F_1
Gold parses + No EP	86.77	86.77	86.77	39.63	39.63	39.63
Gold parses + EP	83.19	82.65	82.92	26.21	27.63	26.90
Automatic parses + EP	81.19	80.04	80.61	24.54	26.45	25.46

Table 2.3: Discourse parser performance summary with or without error propagation (EP) from the upstream components (Lin et al., 2014)

Substantial amount of work in discourse parsing has been done on both on RST and PDTB. The task is to parse a piece of text into the structure imposed by the corresponding theories. In the case of PDTB, we have to first find all of the discourse connectives and then all of the abstract objects that might participate in a discourse relation. Lastly, the system classifies the relation as one of the senses. The PDTB end-to-end discourse parser was first introduced by Lin et al. (2010b) and presents a typical architecture of a discourse parser pursued by subsequent work (Xue et al., 2015; Wang and Lan, 2015; Song et al., 2015). In this architecture, the necessary components are discourse connective detector, argument extractor for explicit discourse relation, argument extractor for implicit discourse relation, discourse connective classifier, and implicit discourse relation sense classifier (Lin et al., 2014), and the parser works in a multistep pipeline (Figure 2.3). The end output of this system is precisely a list of discourse relations as they are annotated in the PDTB. This

system is also significant in that it decomposes the task of discourse parsing into multiple fundamental components, through which we can improve piecewise.

2.2 Modeling Implicit Discourse Relations in the PDTB

The performance bottleneck of the discourse parser is on implicit discourse relations as it is proven to be a much harder task. The F_1 scores for the explicit relations hover around 80, but the F_1 scores for the implicit relations fall flat at around 25, which makes the whole system unusable due to the low overall performance (Table 2.3). If we improve on the implicit discourse relation classifier up to certain point, the discourse parser will show real practical value in the downstream application. In this dissertation, we focus on the implicit discourse relation classifier in particular.

The pioneering work in implicit discourse relation classification relies heavily on specific linguistic lexicon and features derived from syntactic parses (Pitler et al., 2008; Pitler et al., 2009; Lin et al., 2009). The previous work follows the traditional paradigm of crafting linguistically motivated features and loading them up in an off-the-shelf machine learning algorithm such as Naive Bayes or Maximum Entropy Models. Pitler et al. (2009) built the first implicit discourse relation classifier, which employs a battery of discrete features. Most of the features are centered around turning the raw tokens into the their semantic classes. The features are based on various lexicons namely Multi-perspective Question Answering opinion corpus for the semantic polarity, Inquirer tags for the general semantic category, and Levin's verb lexicon for the syntactically motivated verb classes. The Cartesian products are then taken between the features from each argument. The main motivation for using these lexicons is that the discourse relation interpretation requires the semantics of the individual words and how the semantics interact across sentences within the relation, which should

be captured by the Cartesian products. This idea achieves a sizable improvement over the baseline and presents a strong benchmark for later work. In our work, we use these two ideas of semantic representation and inter-argument interaction to inspire the distributed approaches which can better capture the semantics more efficiently and model the inter-argument interaction more compactly.

Another effective feature set is production rules (Lin et al., 2009). The features are generated by taking the Cartesian product of all of the production rules that constitute the parse tree in Arg1 with all of the production rules that constitute the parse tree in Arg2. The same thing can be done on the dependency parses. This feature set is particular useful when careful feature selection based on mutual information is performed (Lin et al., 2009; Ji and Eisenstein, 2015). This feature set makes sense because implicit discourse relation is sensitive to syntactic parallelism and certain fixed structures that are used to signal certain senses without the use of discourse connectives. In other words, the discourse relation can be signaled syntactically and make the use of discourse connective unnatural or unnecessary. This work sheds the light on how syntax might contribute the inference of discourse relations and motivates us to incorporate syntactic structures in the neural network model when composing a feature vector up from the the individual words.

The feature sets above are at best described as a brute-force linguistically-uninspiring solution to the problem and necessitate copious amount of feature selection. Lin et al. (2009) carries out an extensive series of feature selection and combination experiment to sharpen this feature-based approach. The modified definition of mutual information is used to rank features. The best combination consists a few hundreds of features in each category namely word pairs, production rules, and dependency rules. This method is clearly far from obtaining the optimal solution as the problem of feature selection is NP-hard. Samples of random subset of features have been used to improve the performance even further, presenting a

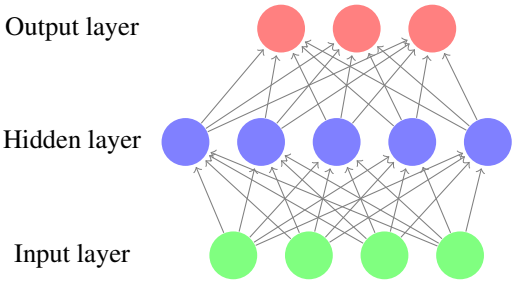

Figure 2.4: Fully connected feedforward neural network architecture with one hidden layer.

brute-force solution to the original brute-force solution (Park and Cardie, 2012). These experimental results confirm our suspicion that many features introduced into the feature space are plain garbage and almost impossible to detect and remove. This is one of the major drawbacks of this surface feature approach especially in the task of implicit discourse relation classification. One of our later models try to tackle this problem directly by bypassing all of the discrete features that flood the feature space to begin with and use the distributed features instead.

2.3 Neural Network Modeling

Another winter of neural networks has come and past. This time around, neural networks are back with a vengeance for many applications of machine learning. The same old model coupled with modern advances in computing has revolutionized the computer vision (LeCun et al., 2010; Krizhevsky et al., 2012), speech recognition (Mohamed et al., 2012), language modeling (Sundermeyer et al., 2012b), and machine-learning paradigms in NLP in general (Socher et al., 2013a; Socher et al., 2012; Devlin et al., 2014). Neural networks have become industry-standard algorithms that achieve as low as 9% word error rate in speech recognition, abandoning the former decade-old set of algorithms that dominated the field (Senior et al., 2015). However, many NLP tasks have been rethought and revisited with neural network

modeling with mixed results, and it is unclear whether neural network is the clear winner in NLP. This question opens door for many research efforts in harnessing the power of neural network in NLP.

Feedforward neural network

The main advantage of neural network comes from the non-linearity, which is difficult to achieve from the standard discriminative models used in NLP and from data-driven feature abstraction. In a vanilla feedforward neural network, we first create a feature vector specifically designed for the task and apply two operations alternately. We apply linear transformation through matrix multiplication and element-wise non-linear transformation through sigmoidal, hyperbolic tangent, or other non-linear functions. These transformations result in another feature vector that can then be used for classification or another round of transformation (Figure 2.4). More precisely, let $X \in \mathbf{R}^d$ be the feature vector we create for the task. Derived features $H_{(1)}$ are created from linear combinations of the inputs:

$$H_{(1)} = \sigma(W_{(1)} \cdot X + b_{(1)})$$

, where $W_{(1)} \in \mathbf{R}^{k \times d}$ is a weight matrix, and $b_{(1)} \in \mathbf{R}^k$ is a bias vector. The (non-linear) activation function $\sigma(\cdot)$ is usually chosen to be tanh or sigmoid function, but other options abound and are still an active area of research. The optional subsequent derived features or hidden layers have the same form:

$$H_{(i)} = \sigma(W_{(i)} \cdot H_{(i-1)} + b_{(i)})$$

A feedforward neural network with t hidden layers models the output posterior probability vector O as a function of the output score vector Y:

$$\begin{aligned} Y &= W_o \cdot H_{(t)} + b_o \\ O &= softmax(Y) \\ &= \frac{\exp(Y)}{\sum_{l=1}^{L} \exp Y_l} \end{aligned}$$

To train a neural network, we have to define the cost function over which we minimize using an optimization algorithm. A popular cost function is cross-entropy loss function (CE). If we have a label categorical variable l^i and a feature vector X^i for each instance i, then the the cross-entropy is the sum of negative label log-likelihood over all N instances in the training set.

$$\begin{aligned} \mathcal{L}_{CE} &= \sum_{i}^{N} \log P(l^i|X^i) \\ &= \sum_{i}^{N} O_{l^i}^i \end{aligned}$$

The loss functions in neural network can only be optimized with a gradient-based method as there is no closed-form solution like the one for some probabilistic models. The gradient is tractable but expensive to compute. The gradient can be computed efficiently using back-propagation algorithm. The algorithm emerges from the fact that we take partial derivative with respect to each parameter in the model. We first compute the gradient with respect to the parameters in the output layer and then backpropagate some of the computation to the earlier layer. Once we compute the gradients for all parameters, we have to use variants of

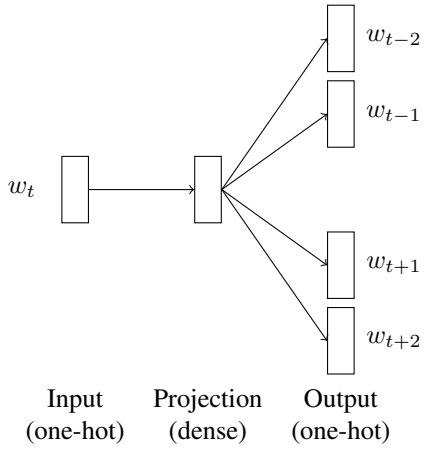

Figure 2.5: In the Skip-gram architecture, word vectors (projection) are induced such that one can use the learned weights to predict the co-occurring words (output).

stochastic gradient methods and other computational "tricks" to speed up and stabilize the learning process such as Adagrad, AdaDelta, or momentum method (Duchi et al., 2011a; Zeiler, 2012; Polyak, 1964).

Word embedding

Besides the algorithmic strength in feature abstraction through hidden layers, neural networks also lend themselves structurally for distributed representation of many linguistic units. The classical paradigms in linguistics often characterize a linguistic unit with a category (e.g. part-of-speech) or a list of categorical features (e.g. a node in a constituent parse). Instead, we can treat a linguistic unit as a vector of continuous-valued features. The success of neural networking in NLP takes its root in word embeddings or word vectors (Mikolov et al., 2013b). Word vectors induced by neural network architectures such as Skip-gram or Continuous Bag-Of-Word architectures are shown to superior to the word vectors induced by counting and reducing dimensionality (Baroni et al., 2014). Figure 2.5 shows the Skip-gram architecture schematically. The Skip-gram architecture posits the objective function

$\mathcal{L}_{skipgram}$ such that the word vector x_t (projected from one-hot vector w_t) predicts the word within the window of size k.

$$\mathcal{L}_{skipgram} = \sum_{t}^{T} \sum_{i=-k;i\neq 0}^{k} \log P(w_{t+k}|x_t)$$

Word vectors can be used as a lexical building block for many downstream tasks as they encode the semantics of the word better than the usual one-hot representation. For example, neural language models have improved substantially over the traditional n-gram language models due to the changes in the representation of a word and the representation of a context (Sundermeyer et al., 2012b; Mikolov et al., 2013a). The categorical representation of a word is replaced by a word vector that can encode some syntactic and semantic properties of the word, and the categorical representation of a context is replaced by a context vector that can encode infinite length of the context.

Furthermore, other linguistic units can also be represented as a vector. Constituents in phrase structure parses have been represented as a vector and used as feature for constituent-level sentiment analysis (Socher et al., 2013b). This representation can also be used for the task of parsing itself (Socher et al., 2013a; Chen and Manning, 2014). With the sequence-to-sequence neural architecture, machine translation has also moved away from the categorical translation tables to neural machine translation with distributed representation of words and phrases (Bahdanau et al., 2014; Cho et al., 2014). Our work uses word vectors induced in this fashion to build up the discourse representation. We come up with multiple neural architectures that utilize the labeled data in the corpus to derive the distributed discourse vector.

Neural discourse parser along with the use of word embedding to derive features has gained some attention within the recent years. A few efforts have been made to create

distributed representation of a discourse unit in conjunction with designing neural network model architecture suitable for the task. Ji and Eisenstein (Ji and Eisenstein, 2014) explore the idea of using word embedding to construct the representation of elementary discourse units (EDU) in parsing the RST discourse structure. They compare bag-of-word features and an EDU vector composed by summing up the word vectors in the EDU. It turns out that bag-of-word features work better. However, they have not explored extensively the effects of different work embeddings and different ways of composing the EDU vector. These two are the core for neural network models that work well and deserve more experimentation. In our work, we conduct series of experiments on the PDTB labels such that we can directly compare different word vectors and discourse vector composition.

A more systematic comparison of various word representation has been conducted for the PDTB discourse relations. Braud and Denis (2015) uses Maximum Entropy model to evaluate the different distributed features on the four binary classification formulation. The feature vector is formed by adding or multiplying all of the word vectors in each of the arguments in the discourse relation. They show that dense distributed features outperform sparser features on some categories. Better results can be achieved when distributed and sparse features are used together and when all of the words (as opposed to just head words) are included in the composition. This suggests to us that distributed features might be the key to further improvement. In our work, we further explore ways of composing feature vectors and exploit the strength of the dense feature vectors in the deeper neural network models.

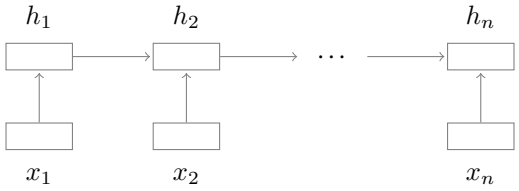

Figure 2.6: Recurrent neural network architecture

Recurrent neural network

Another breakthrough in the recent development of neural network modeling is from the recurrent models. In contrast to the feedforward architecture described earlier, recurrent neural network (RNN) models the influence of the data in a sequence (Hochreiter and Schmidhuber, 1997; Elman, 1990). They can naturally be used for modeling sequences in the similar fashion to Hidden Markov models or linear-chained Conditional Random Fields. The simplest form of RNN, also called Elman Network or "vanilla RNN" , passes on the hidden state or activation to the next time step (Figure 2.6). In other words, the hidden state of the current time step H_t is a function of the input layer of the current time step X_t and the hidden state of the previous time step H_{t-1}. The choice of non-linear activation function is similar to the feedforward net. The hidden activation H_t can be written as a recurrent relation:

$$\begin{aligned} H_t &= g(H_{t-1}, X_t) \\ &= \sigma(W \cdot H_{t-1} + U \cdot X_t + b), \end{aligned}$$

where W and U are weight matrices, and b is a bias vector. The output layer remains the same.

Training recurrent neural network involves backpropagation back-through-time (BPTT) algorithm. BPTT is in essence the same as backpropagation for feedforward net, but the gradient calculation now involves the whole sequence of hidden states and the whole sequence

of outputs. To make the computation more efficient, we can determine at how many time steps the the error from the output layer stops propagating. This approximation does not hurt the performance much when set appropriately, but it is the quick-and-dirty cure to a computational symptom in recurrent neural network.

The multiplicative operations in the vanilla RNN lead to exploding and vanishing gradient problem, which makes it converge too slowly or converge to a suboptimal point. During the training process of RNN, the backpropagated gradient sometimes becomes too small, exerting too little influence back through time. The opposite problem can also occur. If the backpropagated error becomes just slightly large in one time step, it can become exponentially large as it is backpropagated through time and make the training process unfeasible. Long short-term memory (LSTM) network is introduced to address these problems (Hochreiter and Schmidhuber, 1997). LSTM still maintains the recurrent structure in the model, but it differs in how we compute the hidden state H_t. A standard variant of LSTM introduces three gating layers: input gating layer i_t, forget gating layer f_t, and output gating layer o_t. These gating layers are typically a function of the input layer at the current time step and the hidden state at the previous time step.

$$\begin{aligned} i_t &= \text{sigmoid}(W_i \cdot X_t + U_i \cdot H_{t-1} + b_i) \\ f_t &= \text{sigmoid}(W_f \cdot X_t + U_f \cdot H_{t-1} + b_f) \\ o_t &= \text{sigmoid}(W_o \cdot X_t + U_o \cdot H_{t-1} + b_o) \end{aligned}$$

In addition, LSTM introduces the candidate memory cell c'_t and the memory cell c_t. These memory cells help the model control which part of the previous time step should be forgotten or remembered. The memory cells and the gating layers work together to alleviate

the vanishing and exploding gradients.

$$\begin{aligned} c'_t &= \tanh(W_c \cdot X_t + U_c \cdot H_{t-1} + b_c) \\ c_t &= c'_t * i_t + c_{t-1} * f_t \\ H_t &= c_t * o_t \end{aligned}$$

The idea of word embedding and recurrent neural network have changed the way we view sequence modeling and sequence tagging problems in NLP. Both vanilla RNN and LSTM-RNN can be used as effectively as a language model and outperforms the Kneser-Ney and Witten-Bell language models that have dominated as the state-of-the-art for decades (Sundermeyer et al., 2012b). Each word is replaced with its corresponding word embedding, and the hidden state now acts as a context vector that helps predict the next word. The deep variant of RNN has been used to substantially improve the performance of opinion mining (İrsoy and Cardie, 2014). This work formulate the task of opinion mining as a sequence tagging problem. RNNs are stacked up to form a deep RNN, and the features are the word embedding. The deep model and small word embedding perform much better than Conditional Random Fields.

Recurrent neural networks, especially LSTM-RNN, open up the new paradigm for sequence-to-sequence induction such as machine translation and parsing. Machine translation used to rely on discrete translation table. Neural machine translation makes use of word embedding and RNN to create a representation for the source language and prediction of the target language. Bahdanau et al.(2014) uses deep LSTM to encode a sequence of word embeddings in the source language into a single hidden vector. This source language sentence vector is then passed into an LSTM decoder to generate the translation. This similar sequence-to-sequence induction using LSTMs as the encoder and the decoder is also used in parsing (Vinyals et al.,

2015). The source language is the string of text, and the target language is the tree string. These works yield at least two insights for discourse modeling. Word embeddings can be composed through an LSTM-based encoder to create the representation that lends itself for predicting discourse relation. Secondly, the shallow sequential structure in LSTM-RNN is capable of capturing deeper tree structure. These two insights motivate us in using word embeddings as a lexical semantic representation and in using neural network in encoding or composing a feature vector from the word embeddings.

Recurrent neural network has also been used for discourse modeling. Notably, Ji and Eisenstein (2015) have restructured the RNN around a binary syntactic tree. A tree-structured RNN is also called recursive neural network, which has also been used for parsing and sentiment analysis task (Socher et al., 2013a; Socher et al., 2013b; Tai et al., 2015). The hidden state corresponds to an intermediate non-terminal in the phrase structure tree. The root node serves as the feature vector for implicit discourse relation classification. The two feature vectors, each of which comes from an argument, are put through bilinear tensor product, resulting in all interaction terms in the feature vectors. The performance of these features are below the system that uses surface features, but the performance reaches the state-of-the-art level when the recursive neural net and the surface feature system are combined. However, some questions remain as the complexity of this model is not well dissected in the study. It is not clear whether the tree structure is needed, and it is also not clear whether we can better model the inter-argument interaction without bilinear tensor product. In our work, we want to build the neural network model from the ground up and investigate the effects of multiple subcomponents and their contribution to discourse modeling.

Discourse-driven language models have been on the research agenda for speech language modeling and have been shown to be effective when integrated with RNNs (Ji et al., 2016). This particular model takes on the idea that the words in the next sentence should be

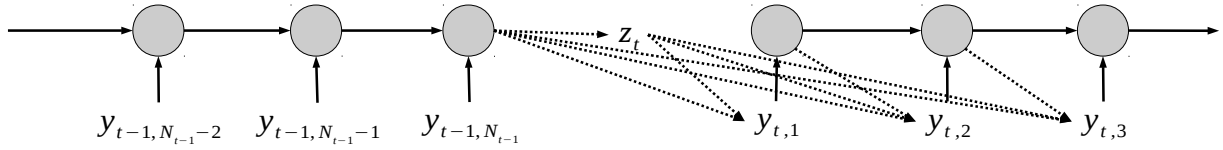

Figure 2.7: Discourse-driven RNN language model. z denotes the discourse relation sense between the two sentences. y denotes words in sentences.

influenced by the words in the previous sentence and also by the discourse relation. Ji et al. (2016) combine RNN language model and discriminative output layer that predicts the discourse relation sense (Figure 2.7). This approach works well in the top four-way classification, but it is unclear whether it can adapt to different label sets. If such discourse parser performs k-way classification, it has to split the data into k separate parts when fitting language model parameters although they do share some of the parameters. The discourse data is almost always small because of the annotation cost, and the label set varies from tasks to tasks. Therefore, we propose a model that is robust against varying label sets and small data set size, characteristic to discourse annotation data.

Chapter 3

Brown Cluster Pair and Coreferential Pattern Features

This chapter is adapted from Rutherford and Xue (2014).

One of the key aspects for discourse relation modeling is the semantic interaction between the two text spans. A good key word pair should be able to signal the type of relationship that holds between the two text spans. For example, *Carl gets hungry. He walks swiftly to the pantry.*. The inter-argument word pair *hungry-pantry* signals the likely sense of causal relation without considering the rest of the sentences. This simple inter-argument semantic interaction should be able to guide a machine-learning system to identify the sense of the discourse relation.

Existing systems for implicit discourse relation classification, which make heavy use of word pairs, suffer from data sparsity problem. There is no good way to filtering which word pairs to be included as features, which results in a large feature set. Additionally, a word pair in the training data may not appear in the test data, so the generalization can be quite low. A better representation of two adjacent sentences beyond word pairs could have a

significant impact on predicting the sense of the discourse relation that holds between them. Data-driven theory-independent word classification such as Brown clustering should be able to provide a more compact word representation (Brown et al., 1992). Brown clustering algorithm induces a hierarchy of words in a large unannotated corpus based on word co-occurrences within the window. The induced hierarchy might give rise to features that we would otherwise miss. In this chapter, we propose to use the cartesian product of Brown cluster assignment of the sentence pair as an alternative abstract word representation for building an implicit discourse relation classifier.

Through word-level semantic commonalities revealed by Brown clusters and entity-level relations revealed by coreference resolution, we might be able to paint a more complete picture of the discourse relation in question. Coreference resolution unveils the patterns of entity realization within the discourse, which might provide clues for the types of the discourse relations. The information about certain entities or mentions in one sentence should be carried over to the next sentence to form a coherent relation. It is possible that coreference chains and semantically-related predicates in the local context might show some patterns that characterize types of discourse relations. We hypothesize that coreferential rates and coreference patterns created by Brown clusters should help characterize different types of discourse relations.

Here, we introduce two novel sets of features for implicit discourse relation classification. Further, we investigate the effects of using Brown clusters as an alternative word representation and analyze the impactful features that arise from Brown cluster pairs. We also study coreferential patterns in different types of discourse relations in addition to using them to boost the performance of our classifier. These two sets of features along with previously used features outperform the baseline systems by approximately 5% absolute across all categories and reveal many important characteristics of implicit discourse relations.

	Number of instances	
	Implicit	Explicit
Comparison	2503 (15.11%)	5589 (33.73%)
Contingency	4255 (25.68%)	3741 (22.58%)
Expansion	8861 (53.48%)	72 (0.43%)
Temporal	950 (5.73%)	3684 (33.73%)
Total	16569 (100%)	13086 (100%)

Table 3.1: The distribution of senses of implicit discourse relations is imbalanced.

3.1 Sense annotation in Penn Discourse Treebank

The Penn Discourse Treebank (PDTB) is the largest corpus richly annotated with explicit and implicit discourse relations and their senses (Prasad et al., 2008). PDTB is drawn from Wall Street Journal articles with overlapping annotations with the Penn Treebank (Marcus et al., 1993). Each discourse relation contains the information about the extent of the arguments, which can be a sentence, a constituent, or an incontiguous span of text. Each discourse relation is also annotated with the sense of the relation that holds between the two arguments. In the case of implicit discourse relations, where the discourse connectives are absent, the most appropriate connective is annotated.

The senses are organized hierarchically. Our focus is on the top level senses because they are the four fundamental discourse relations that various discourse analytic theories seem to converge on (Mann and Thompson, 1988). The top level senses are Comparison, Contingency, Expansion, and Temporal.

The explicit and implicit discourse relations almost orthogonally differ in their distributions of senses (Table 3.1). This difference has a few implications for studying implicit discourse relations and uses of discourse connectives (Patterson and Kehler, 2013). For example, Temporal relations constitute only 5% of the implicit relations but 33% of the explicit relations because they might not be as natural to create without discourse connec-

tives. On the other hand, EXPANSION relations might be more cleanly achieved without ones as indicated by its dominance in the implicit discourse relations. This imbalance in class distribution requires greater care in building statistical classifiers (Wang et al., 2012).

3.2 Experiment setup

We followed the setup of the previous studies for a fair comparison with the two baseline systems by Pitler et al. (2009) and Park and Cardie (2012). The task is formulated as four separate one-against-all binary classification problems: one for each top level sense of implicit discourse relations. In addition, we add one more classification task with which to test the system. We merge ENTREL with EXPANSION relations to follow the setup used by the two baseline systems. An argument pair is annotated with ENTREL in PDTB if an entity-based coherence and no other type of relation can be identified between the two arguments in the pair. In this study, we assume that the gold standard argument pairs are provided for each relation. Most argument pairs for implicit discourse relations are a pair of adjacent sentences or adjacent clauses separated by a semicolon and should be easily extracted.

The PDTB corpus is split into a training set, development set, and test set the same way as in the baseline systems. Sections 2 to 20 are used to train classifiers. Sections 0–1 are used for developing feature sets and tuning models. Section 21–22 are used for testing the systems.

The statistical models in the following experiments are from MALLET implementation (McCallum, 2002) and libSVM (Chang and Lin, 2011). For all five binary classification tasks, we try Balanced Winnow (Littlestone, 1988), Maximum Entropy, Naive Bayes, and Support Vector Machine. The parameters and the hyperparameters of each classifier are set to their default values. The code for our model along with the data matrices is available at

`github.com/attapol/brown_coref_implicit`.

3.3 Feature Description

Unlike the baseline systems, all of the features in the experiments use the output from automatic natural language processing tools. We use the Stanford CoreNLP suite to lemmatize and part-of-speech tag each word (Toutanova et al., 2003; Toutanova and Manning, 2000), obtain the phrase structure and dependency parses for each sentence (De Marneffe et al., 2006; Klein and Manning, 2003), identify all named entities (Finkel et al., 2005), and resolve coreference (Raghunathan et al., 2010; Lee et al., 2011; Lee et al., 2013).

3.3.1 Features used in previous work

The baseline features consist of the following: First, last, and first 3 words, numerical expressions, time expressions, average verb phrase length, modality, General Inquirer tags, polarity, Levin verb classes, and production rules. These features are described in greater detail by Pitler et al. (2009).

3.3.2 Brown cluster pair features

To generate Brown cluster assignment pair features, we replace each word with its hard Brown cluster assignment. We used the Brown word clusters provided by MetaOptimize (Turian et al., 2010). 3,200 clusters were induced from RCV1 corpus, which contains about 63 million tokens from Reuters English newswire. Then we take the Cartesian product of the Brown cluster assignments of the words in Arg1 and the ones of the words in Arg2. For example, suppose Arg1 has two words $w_{1,1}, w_{1,2}$, Arg2 has three words $w_{2,1}, w_{2,2}, w_{2,3}$,

and then $B(.)$ maps a word to its Brown cluster assignment. A word w_{ij} is replaced by its corresponding Brown cluster assignment $b_{ij} = B(w_{ij})$. The resulting word pair features are $(b_{1,1}, b_{2,1}), (b_{1,1}, b_{2,2}), (b_{1,1}, b_{2,3}), (b_{1,2}, b_{2,1}), (b_{1,2}, b_{2,2})$, and $(b_{1,2}, b_{2,3})$.

Therefore, this feature set can generate $O(3200^2)$ binary features. The feature set size is orders of magnitude smaller than using the actual words, which can generate $O(V^2)$ distinct binary features where V is the size of the vocabulary.

3.3.3 Coreference-based features

We want to take advantage of the semantics of the sentence pairs even more by considering how coreferential entities play out in the sentence pairs. We consider various inter-sentential coreference patterns to include as features and also to better describe each type of discourse relation with respect to its place in the coreference chain.

For compactness in explaining the following features, we define *similar words* to be the words assigned to the same Brown cluster.

Number of coreferential pairs: We count the number of inter-sentential coreferential pairs. We expect that EXPANSION relations should be more likely to have coreferential pairs because the detail or information about an entity mentioned in Arg1 should be expanded in Arg2. Therefore, entity sharing might be difficult to avoid.

Similar nouns and verbs: A binary feature indicating whether similar or coreferential nouns are the arguments of the similar predicates. Predicates and arguments are identified by dependency parses. We notice that sometimes the author uses synonyms while trying to expand on the previous predicates or entities. The words that indicate the common topics might be paraphrased, so exact string matching cannot detect whether the two arguments still focus on the same topic. This might be useful for identifying CONTINGENCY relations as

they usually discuss two causally-related events that involve two seemingly unrelated agents and/or predicates.

Similar subject or main predicates: A binary feature indicating whether the main verbs of the two arguments have the same subjects or not and another binary feature indicating whether the main verbs are similar or not. For our purposes, the two subjects are said to be the same if they are coreferential or assigned to the same Brown cluster. We notice that COMPARISON relations usually have different subjects for the same main verbs and that TEMPORAL relations usually have the same subjects but different main verbs.

	Current			Park and Cardie (2012)	Pitler et al. (2009)
	P	R	F_1	F_1	F_1
COMPARISON vs others	27.34	72.41	**39.70**	31.32	21.96
CONTINGENCY vs others	44.52	69.96	**54.42**	49.82	47.13
EXPANSION vs others	59.59	85.50	70.23	-	-
EXP+ENTREL vs others	69.26	95.92	**80.44**	79.22	76.42
TEMPORAL vs others	18.52	63.64	**28.69**	26.57	16.76

Table 3.2: Our classifier outperform the previous systems across all four tasks without the use of gold-standard parses and coreference resolution.

3.3.4 Feature selection and training sample reweighting

The nature of the task and the dataset poses at least two problems in creating a classifier. First, the classification task requires a large number of features, some of which are too rare and inconducive to parameter estimation. Second, the label distribution is highly imbalanced (Table 3.1) and this might degrade the performance of the classifiers (Japkowicz, 2000). Recently, Park and Cardie (2012) and Wang et al. (2012) addressed these problems directly by optimally select a subset of features and training samples. Unlike previous work, we do not discard any of data in the training set to balance the label distribution. Instead,

Comparison

Feature set	F_1	% change
All features	39.70	-
All excluding Brown cluster pairs	35.71	-10.05%
All excluding Production rules	37.27	-6.80%
All excluding First, last, and First 3	39.18	-1.40%
All excluding Polarity	39.39	-0.79%

Contingency

Feature set	F_1	% change
All	54.42	-
All excluding Brown cluster pairs	51.50	-5.37%
All excluding First, last, and First 3	53.56	-1.58%
All excluding Polarity	53.82	-1.10%
All excluding Coreference	53.92	-0.92%

Expansion

Feature set	F_1	% change
All	70.23	-
All excluding Brown cluster pairs	67.48	-3.92%
All excluding First, last, and First 3	69.43	-1.14%
All excluding Inquirer tags	69.73	-0.71%
All excluding Polarity	69.92	-0.44%

Temporal

Feature set	F_1	% change
All	28.69	-
All excluding Brown cluster pairs	24.53	-14.50%
All excluding Production rules	26.51	-7.60%
All excluding First, last, and First 3	26.56	-7.42%
All excluding Polarity	27.42	-4.43%

Table 3.3: Ablation study: The four most impactful feature classes and their relative percentage changes are shown. Brown cluster pair features are the most impactful across all relation types.

we reweight the training samples in each class during parameter estimation such that the performance on the development set is maximized. In addition, the number of occurrences for each feature must be greater than a cut-off, which is also tuned on the development set

to yield the highest performance on the development set.

3.4 Results

Our experiments show that the Brown cluster and coreference features along with the features from the baseline systems improve the performance for all discourse relations (Table 3.2). Consistent with the results from previous work, the Naive Bayes classifier outperforms MaxEnt, Balanced Winnow, and Support Vector Machine across all tasks regardless of feature pruning criteria and training sample reweighting. A possible explanation is that the small dataset size in comparison with the large number of features might favor a generative model like Naive Bayes (Jordan and Ng, 2002). So we only report the performance from the Naive Bayes classifiers.

It is noteworthy that the baseline systems use the gold standard parses provided by the Penn Treebank, but ours does not because we would like to see how our system performs realistically in conjunction with other pre-processing tasks such as lemmatization, parsing, and coreference resolution. Nevertheless, our system still manages to outperform the baseline systems in all relations by a sizable margin.

Our preliminary results on implicit sense classification suggest that the Brown cluster word representation and coreference patterns might be indicative of the senses of the discourse relations, but we would like to know the extent of the impact of these novel feature sets when used in conjunction with other features. To this aim, we conduct an ablation study, where we exclude one of the feature sets at a time and then test the resulting classifier on the test set. We then rank each feature set by the relative percentage change in F_1 score when excluded from the classifier. The data split and experimental setup are identical to the ones described in the previous section but only with Naive Bayes classifiers.

The ablation study results imply that Brown cluster features are the most impactful feature set across all four types of implicit discourse relations. When ablated, Brown cluster features degrade the performance by the largest percentage compared to the other feature sets regardless of the relation types(Table 3.3). TEMPORAL relations benefit the most from Brown cluster features. Without them, the F_1 score drops by 4.12 absolute or 14.50% relative to the system that uses all of the features.

3.5 Feature analysis

3.5.1 Brown cluster features

This feature set is inspired by the word pair features, which are known for its effectiveness in predicting senses of discourse relations between the two arguments. Marcu et al (2002a), for instance, artificially generated the implicit discourse relations and used word pair features to perform the classification tasks. Those word pair features work well in this case because their artificially generated dataset is an order of magnitude larger than PDTB. Ideally, we would want to use the word pair features instead of word cluster features if we have enough data to fit the parameters. Consequently, other less sparse handcrafted features prove to be more effective than word pair features for the PDTB data (Pitler et al., 2009). We remedy the sparsity problem by clustering the words that are distributionally similar together and greatly reduce the number of features.

Since the ablation study is not fine-grained enough to spotlight the effectiveness of the individual features, we quantify the predictiveness of each feature by its mutual information. Under Naive Bayes conditional independence assumption, the mutual information between the features and the labels can be efficiently computed in a pairwise fashion. The mutual

information between a binary feature X_i and class label Y is defined as:

$$I(X_i, Y) = \sum_{y} \sum_{x=0,1} \hat{p}(x, y) \log \frac{\hat{p}(x, y)}{\hat{p}(x)\hat{p}(y)}$$

$\hat{p}(\cdot)$ is the probability distribution function whose parameters are maximum likelihood estimates from the training set. We compute mutual information for all four one-vs-all classification tasks. The computation is done as part of the training pipeline in MALLET to ensure consistency in parameter estimation and smoothing techniques. We then rank the cluster pair features by mutual information. The results are compactly summarized in bipartite graphs shown in Figure 3.1, where each edge represents a cluster pair. Since mutual information itself does not indicate whether a feature is favored by one or the other label, we also verify the direction of the effects of each of the features included in the following analysis by comparing the class conditional parameters in the Naive Bayes model.

The most dominant features for COMPARISON classification are the pairs whose members are from the same Brown clusters. We can distinctly see this pattern from the bipartite graph because the nodes on each side are sorted alphabetically. The graph shows many parallel short edges, which suggest that many informative pairs consist of the same clusters. Some of the clusters that participate in such pair consist of named-entities from various categories such as airlines (*King, Bell, Virgin, Continental, ...*), and companies (*Thomson, Volkswagen, Telstra, Siemens*). Some of the pairs form a broad category such as political agents (*citizens, pilots, nationals, taxpayers*) and industries (*power, insurance, mining*). These parallel patterns in the graph demonstrate that implicit COMPARISON relations might be mainly characterized by juxtaposing and explicitly contrasting two different entities in two adjacent sentences.

Without the use of a named-entity recognition system, these Brown cluster pair features

effectively act as features that detect whether the two arguments in the relation contain named-entities or nouns from the same categories or not. These more subtle named-entity-related features are cleanly discovered through replacing words with their data-driven Brown clusters without the need for additional layers of pre-processing.

If the words in one cluster semantically relates to the words in another cluster, the two clusters are more likely to become informative features for CONTINGENCY classification. For instance, technical terms in stock and trading (*weighted, Nikkei, composite, diffusion*) pair up with economic terms (*Trading, Interest, Demand, Production*). The cluster with *analysts* and *pundits* pairs up with the one that predominantly contains quantifiers (*actual, exact, ultimate, aggregate*). In addition to this pattern, we observed the same parallel pair pattern we found in COMPARISON classification. These results suggest that in establishing a CONTINGENCY relation implicitly the author might shape the sentences such that they have semantically related words if they do not mention named-entities of the same category.

Through Brown cluster pairs, we obtain features that detect a shift between generality and specificity within the scope of the relation. For example, a cluster with industrial categories (*Electric, Motor, Life, Chemical, Automotive*) couples with specific brands or companies (*GM, Ford, Barrick, Anglo*). Or such a pair might simply reflects a shift in plurality e.g. *businesses - business* and *Analysts -analyst*. EXPANSION relations capture relations in which one argument provides a specification of the previous and relations in which one argument provides a generalization of the other. Thus, these shift detection features could help distinguish EXPANSION relations.

We found a few common coreference patterns of names in written English to be useful. First and last name are used in the first sentence to refer to a person who just enters the discourse. That person is referred to just by his/her title and last name in the following sentence. This pattern is found to be informative for EXPANSION relations. For example,

the edges (not shown in the graph due to lack of space) from the first name clusters to the title (*Mr, Mother, Judge, Dr*) cluster.

Time expressions constitutes the majority of the nodes in the bipartite graph for TEMPORAL relations. More strikingly, the specific dates (e.g. clusters that have positive integers smaller than 31) are more frequently found in Arg2 than Arg1 in implicit TEMPORAL relations. It is possible that TEMPORAL relations are more naturally expressed without a discourse connective if a time point is clearly specified in Arg2 but not in Arg1.

TEMPORAL relations might also be implicitly inferred through detecting a shift in quantities. We notice that clusters whose words indicate changes e.g. *increase, rose, loss* pair with number clusters. Sentences in which such pairs participate might be part of a narrative or a report where one expects a change over time. These changes conveyed by the sentences constitute a natural sequence of events that are temporally related but might not need explicit temporal expressions.

3.5.2 Coreference features

Coreference features are very effective given that they constitute a very small set compared to the other feature sets. In particular, excluding them from the model reduces F_1 scores for TEMPORAL and CONTINGENCY relations by approximately 1% relative to the system that uses all of the features. We found that the sentence pairs in these two types of relations have distinctive coreference patterns.

We count the number of pairs of arguments that are linked by a coreference chain for each type of relation. The coreference chains used in this study are detected automatically from the training set through Stanford CoreNLP suite (Raghunathan et al., 2010; Lee et al.,

2011; Lee et al., 2013). TEMPORAL relations have a significantly higher coreferential rate than the other three relations ($p < 0.05$, pair-wise t-test corrected for multiple comparisons). The differences between COMPARISON, CONTINGENCY, and EXPANSION, however, are not statistically significant (Figure 3.2).

The choice to use or not to use a discourse connective is strongly motivated by linguistic features at the discourse levels (Patterson and Kehler, 2013). Additionally, it is very uncommon to have temporally-related sentences without using explicit discourse connectives. The difference in coreference patterns might be one of the factors that influence the choice of using a discourse connective to signal a TEMPORAL relation. If sentences are coreferentially linked, then it might be more natural to drop a discourse connective because the temporal ordering can be easily inferred without it. For example,

(3) Her story is partly one of personal downfall. [*previously*] She was an unstinting teacher who won laurels and inspired students... (WSJ0044)

The coreference chain between the two temporally-related sentences in (1) can easily be detected. Inserting *previously* as suggested by the annotation from the PDTB corpus does not add to the temporal coherence of the sentences and may be deemed unnecessary. But the presence of coreferential link alone might bias the inference toward TEMPORAL relation while CONTINGENCY might also be inferred.

Additionally, we count the number of pairs of arguments whose grammatical subjects are linked by a coreference chain to reveal the syntactic-coreferential patterns in different relation types. Although this specific pattern seems rare, more than eight percent of all relations have coreferential grammatical subjects. We observe the same statistically significant differences between TEMPORAL relations and the other three types of relations. More interestingly, the subject coreferential rate for CONTINGENCY relations is the lowest among the three

categories ($p < 0.05$, pair-wise t-test corrected for multiple comparisons).

It is possible that coreferential subject patterns suggest temporal coherence between the two sentences without using an explicit discourse connective. CONTINGENCY relations, which can only indicate causal relationships when realized implicitly, impose the temporal ordering of events in the arguments; i.e. if Arg1 is causally related to Arg2, then the event described in Arg1 must temporally precede the one in Arg2. Therefore, CONTINGENCY and TEMPORAL can be highly confusable. To understand why this pattern might help distinguish these two types of relations, consider these examples:

(4) He also asserted that exact questions weren't replicated. [*Then*] When referred to the questions that match, he said it was coincidental. (WSJ0045)

(5) He also asserted that exact questions weren't replicated. When referred to the questions that match, she said it was coincidental.

When we switch out the coreferential subject for an arbitrary uncoreferential pronoun as we do in (3), we are more inclined to classify the relation as CONTINGENCY.

3.6 Related work

Word-pair features are known to work very well in predicting senses of discourse relations in an artificially generated corpus (Marcu and Echihabi, 2002a). But when used with a realistic corpus, model parameter estimation suffers from data sparsity problem due to the small dataset size. Biran and McKeown (2013) attempts to solve this problem by aggregating word pairs and estimating weights from an unannotated corpus but only with limited success.

Recent efforts have focused on introducing meaning abstraction and semantic representation between the words in the sentence pair. Pitler et al. (2009) uses external lexicons to

replace the one-hot word representation with semantic information such as word polarity and various verb classification based on specific theories (Stone et al., 1968; Levin, 1993). Park and Cardie (2012) selects an optimal subset of these features and establishes the strongest baseline to best of our knowledge.

Brown word clusters are hierarchical clusters induced by frequency of co-occurrences with other words (Brown et al., 1992). The strength of this word class induction method is that the words that are classified to the same clusters usually make an interpretable lexical class by the virtue of their distributional properties. This word representation has been used successfully to augment the performance of many NLP systems (Ritter et al., 2011; Turian et al., 2010).

In addition, Louis et al. (2010) uses multiple aspects of coreference as features to classify implicit discourse relations without much success while suggesting many aspects that are worth exploring. In a corpus study by Louis and Nenkova (2010), coreferential rates alone cannot explain all of the relations, and more complex coreference patterns have to be considered.

3.7 Conclusions

We present statistical classifiers for identifying senses of implicit discourse relations and introduce novel feature sets that exploit distributional similarity and coreference information. Our classifiers outperform the classifiers from previous work in all types of implicit discourse relations. Altogether these results present a stronger baseline for the future research endeavors in implicit discourse relations.

In addition to enhancing the performance of the classifier, Brown word cluster pair features disclose some of the new aspects of implicit discourse relations. The feature analysis

confirms our hypothesis that cluster pair features work well because they encapsulate relevant word classes which constitute more complex informative features such as named-entity pairs of the same categories, semantically-related pairs, and pairs that indicate specificity-generality shift. At the discourse level, Brown clustering is superior to a one-hot word representation for identifying inter-sentential patterns and the interactions between words.

Coreference chains that traverse through the discourse in the text shed the light on different types of relations. The preliminary analysis shows that TEMPORAL relations have much higher inter-argument coreferential rates than the other three senses of relations. Focusing on only subject-coreferential rates, we observe that CONTINGENCY relations show the lowest coreferential rate. The coreference patterns differ substantially and meaningfully across discourse relations and deserve further exploration.

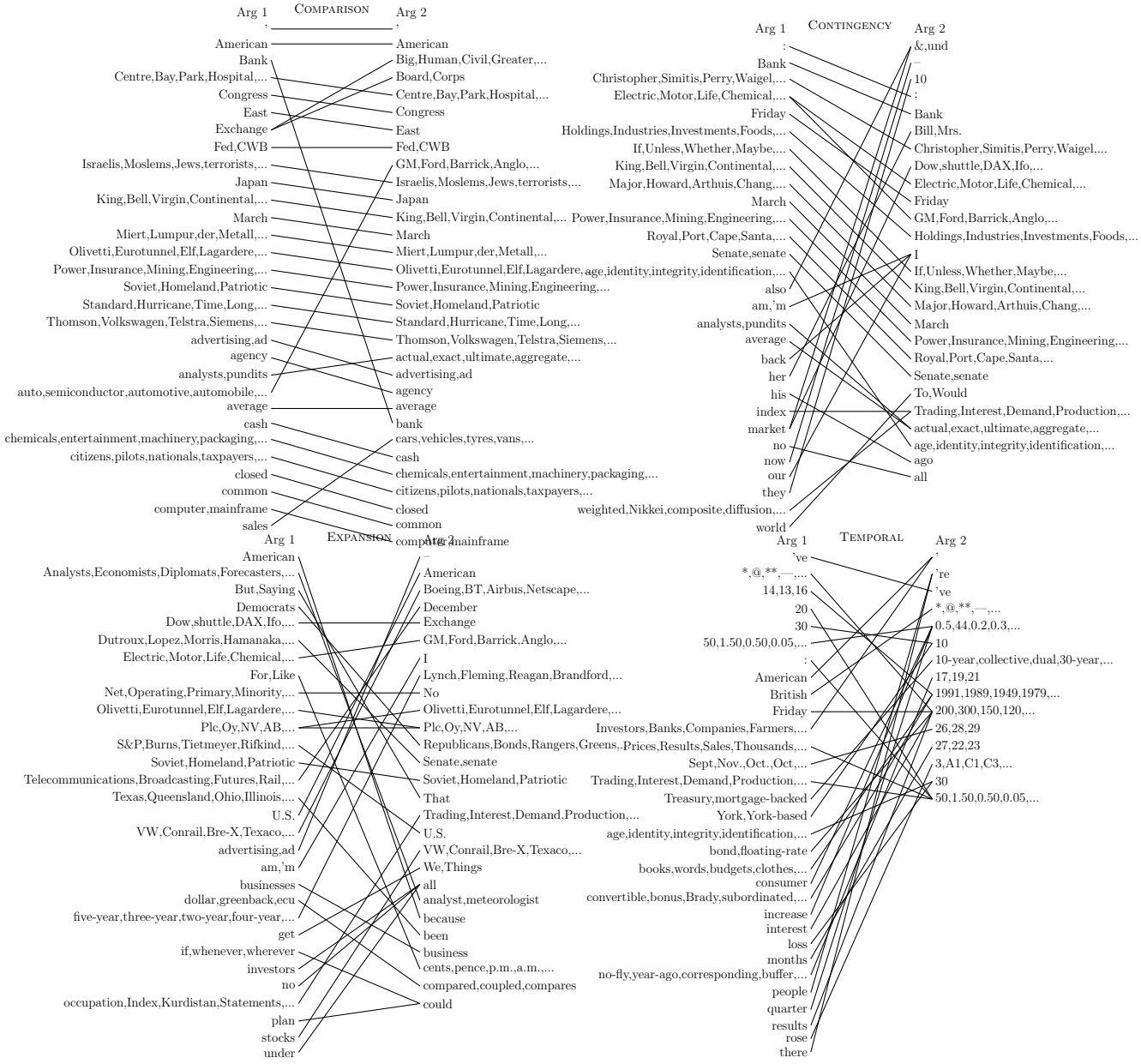

Figure 3.1: The bipartite graphs show the top 40 non-stopword Brown cluster pair features for all four classification tasks. Each node on the left and on the right represents word cluster from Arg1 and Arg2 respectively. We only show the clusters that appear fewer than six times in the top 3,000 pairs to exclude stopwords. Although the four tasks are interrelated, some of the highest mutual information features vary substantially across tasks.

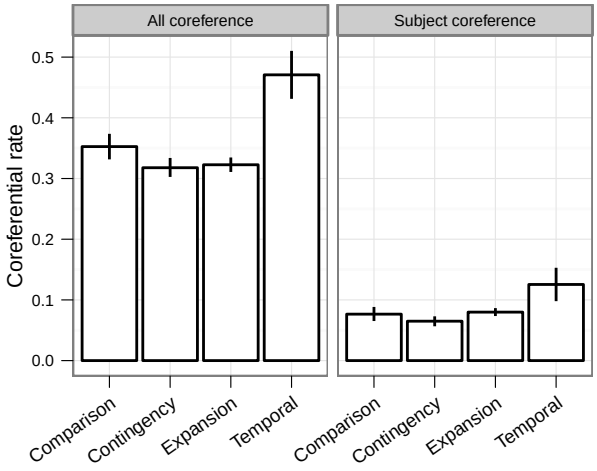

Figure 3.2: The coreferential rate for TEMPORAL relations is significantly higher than the other three relations ($p < 0.05$, corrected for multiple comparison).

Chapter 4

Distant Supervision from Discourse Connectives

This chapter is adapted from Rutherford and Xue (2015).

According to the view of the Penn Discourse Treebank (PDTB) (Prasad et al., 2008), the sense of discourse relation is lexically grounded to the explicit or the omitted underlying discourse connectives. The relations grounded to the explicit (unomitted) discourse relations are called explicit discourse relations. The ones grounded to the omitted discourse relations are called implicit discourse relations. Discourse connectives are a linguistic device that gives a clear signal of the sense of discourse relation. This is the property that we would like to exploit in this chapter. For example,

(6) [The city's Campaign Finance Board has refused to pay Mr Dinkins $95,142 in matching funds]$_{Arg1}$ <u>because</u> [his campaign records are incomplete]$_{Arg2}$.

(7) [So much of the stuff poured into its Austin, Texas, offices that its mail rooms there simply stopped delivering it]$_{Arg1}$. <u>Implicit=so</u> [Now, thousands of mailers, catalogs and

sales pitches go straight into the trash]$_{Arg2}$.

Determining the sense of an explicit discourse relation such as (1) is straightforward since "because" is a strong indicator that the relation between the two arguments is CONTINGENCY.CAUSE. This task effectively amounts to disambiguating the sense of discourse connective, which can be done with high accuracy (Pitler et al., 2008).

However, in the absence of an explicit discourse connective, inferring the sense of a discourse relation has proved to a very challenging task (Park and Cardie, 2012; Rutherford and Xue, 2014). The sense is no longer localized on one or two discourse connectives and must now be inferred solely based on its two textual arguments. Given the limited amount of annotated data in comparison to the number of features needed, the process of building a classifier is plagued by the data sparsity problem (Li and Nenkova, 2014). As a result, the classification accuracy of implicit discourse relations remains much lower than that of explicit discourse relations (Pitler et al., 2008).

Sophisticated feature aggregation and selection methods seem promising but were shown to improve the performance only marginally (Biran and McKeown, 2013; Park and Cardie, 2012). Given that the arguments are typically large linguistic units such as clauses and sentences, we cannot realistically expect repetitions of these linguistic units in a training corpus like we do for individual words or short n-grams in a corpus the size of the PDTB. The standard approach in existing work on inferring implicit discourse relations is to decompose the two arguments into word pairs and use them as features. Since the two arguments for the same type of discourse relation can vary tremendously, a lot of features are needed for a classifier to work well (Park and Cardie, 2012; Rutherford and Xue, 2014).

One potential method for reducing the data sparsity problem is through a distantly supervised learning paradigm, which is the direction we take in this work. Distant supervision

approaches make use of prior knowledge or heuristics to cheaply obtain *weakly labeled data*, which potentially contain a small number of false labels. Weakly labeled data can be collected from unannotated data and incorporated in the model training process to supplement manually labeled data. This approach has recently seen some success in natural language processing tasks such as relation extraction (Mintz et al., 2009), event extraction (Reschke et al., 2014), and text classification (Thamrongrattanarit et al., 2013). For our task, we can collect instances of explicit discourse relations from unannotated data by some simple heuristics. After dropping the discourse connectives, we should be able to treat them as additional implicit discourse relations.

The approach assumes that when the discourse connective is omitted, the discourse relation remains the same, which is a popular assumption in discourse analysis (Fraser, 2006; Schourup, 1999). This assumption turns out to be too strong in many cases as illustrated in (8):

(8) [I want to go home for the holiday]$_{Arg1}$. Nonetheless, [I will book a flight to Hawaii]$_{Arg2}$.

If "Nonetheless" is dropped in (8), one can no longer infer the COMPARISON relation. Instead, one would naturally infer a CONTINGENCY relation. Dropping the connective and adding the relation as a training sample adds noise to the training set and can only hurt the performance. In addition, certain types of explicit discourse relations have no corresponding implicit discourse relations. For example, discourse relations of the type CONTINGENY.CONDITION are almost always expressed with an explicit discourse connective and do not exist in implicit relations. We believe this also explains the lack of success in previous attempts to boost the performance of implicit discourse relation detection with this approach. (Biran and McKeown, 2013; Pitler et al., 2009). This suggests that in order for this approach to work, we need to identify instances of explicit discourse relations that closely match the

characteristics of implicit discourse relations.

In this paper, we propose two criteria for selecting such explicit discourse relation instances: *omission rate* and *context differential*. Our selection criteria first classify discourse connectives by their distributional properties and suggest that not all discourse connectives are truly optional and not all implicit and explicit discourse relations are equivalent, contrary to commonly held beliefs in previous studies of discourse connectives. We show that only the freely omissible discourse connectives gather additional training instances that lead to significant performance gain against a strong baseline. Our approach improves the performance of implicit discourse relations without additional feature engineering in many settings and opens doors to more sophisticated models that require more training data.

The rest of the paper is structured as follows. In Section 4.1, we describe the discourse connective selection criteria. In Section 4.2, we present our discourse connective classification method and experimental results that demonstrate its impact on inferring implicit discourse relations. We discuss related work and conclude our findings in Section 4.3 and 4.4 respectively.

4.1 Discourse Connective Classification and Discourse Relation Extraction

4.1.1 Datasets used for selection

We use two datasets for the purposes of extracting and selecting weakly labeled explicit discourse relation instances: the Penn Discourse Treebank 2.0 (Prasad et al., 2008) and the English Gigaword corpus version 3 (Graff et al., 2007).

The Penn Discourse Treebank (PDTB) is the largest manually annotated corpus of dis-

course relations on top of one million word tokens from the Wall Street Journal (Prasad et al., 2008; Prasad et al., 2007). Each discourse relation in the PDTB is annotated with a semantic sense in the PDTB sense hierarchy, which has three levels: CLASS, TYPE and SUBTYPE. In this work, we are primarily concerned with the four top-level CLASS senses: EXPANSION, COMPARISON, CONTINGENCY, and TEMPORAL. The distribution of top-level senses of implicit discourse relations is shown in Table 4.2. The spans of text that participate in the discourse relation are also explicitly annotated. These are called ARG1 or ARG2, depending on its relationship with the discourse connective.

The PDTB is our corpus of choice for its lexical groundedness. The existence of a discourse relation must be linked or grounded to a discourse connective. More importantly, this applies to not only explicit discourse connectives that occur naturally as part of the text but also to implicit discourse relations where a discourse connective is added by annotators during the annotation process. This is crucial to the work reported here in that it allows us to compare the distribution of the same connective in explicit and implicit discourse relations. In the next subsection, we will explain in detail how we compute the comparison measures and apply them to the selection of explicit discourse connectives that can be used for collecting good weakly labeled data.

We use the Gigaword corpus, a large unannotated newswire corpus, to extract and select instances of explicit discourse discourse relations to supplement the manually annotated instances from the PDTB. The Gigaword corpus is used for its large size of 2.9 billion words and its similarity to the Wall Street Journal data from the PDTB. The source of the corpus is drawn from six distinct international sources of English newswire dating from 1994 - 2006. We use this corpus to extract weakly labeled data for the experiment.

4.1.2 Discourse relation extraction pattern

We extract instances of explicit discourse relations from the Gigaword Corpus that have the same patterns as the implicit discourse relations in the PDTB, using simple regular expressions. We first sentence-segment the Gigaword Corpus using the NLTK sentence segmenter (Bird, 2006). We then write a set of rules to prevent some common erroneous cases such as *because* vs *because of* from being included.

If a discourse connective is a subordinating conjunction, then we use the following pattern:

```
(Clause 1) (connective) (clause 2).
```

Clause 1 and capitalized clause 2 are then used as *Arg1* and *Arg2* respectively.

If a discourse connective is a coordinating conjunction or discourse adverbial, we use the following pattern:

```
(Sentence 1). (Connective),(clause 2).
```

Sentence 1 and Clause 2 with the first word capitalized are used as *Arg1* and *Arg2* respectively.

Although there are obviously many other syntactic patterns associated with explicit discourse connectives, we use these two patterns because these are the only patterns that are also observed in the implicit discourse relations. We want to select instances of explicit discourse relations that match the argument patterns of implicit discourse relations as much as possible. As restrictive as this may seem, these two patterns along with the set of rules allow us to extract more than 200,000 relation instances from the Gigaword corpus, so the coverage is not an issue.

4.1.3 Discourse connective selection and classification criteria

We hypothesize that connectives that are omitted often and in a way that is insensitive to the semantic context are our ideal candidates for extracting good weakly labeled data. We call this

type of connectives *freely omissible discourse connectives*. To search for this class of connectives, we need to characterize connectives by the rate at which they are omitted and by the similarity between their context, in this case their arguments, in explicit and implicit discourse relations. This is possible because implicit discourse connectives are inserted during annotation in the PDTB. For each discourse connective, we can compute *omission rate* and *context differential* from annotated explicit and implicit discourse relation instances in the PDTB and use those measures to classify and select discourse connectives.

Omission rate (OR)

We use *omission rates* (OR) to measure the level of optionality of a discourse connective. The omission rate of a type of discourse connective (DC) is defined as:

$$\frac{\text{\# occurrences of DC in implicit relations}}{\text{\# total occurrences of DC}}$$

Our intuition is that the discourse connectives that have a high level of omission rate are more suitable as supplemental training data to infer the sense of implicit discourse relations.

Context differential

The omission of a freely omissible discourse connective should also be context-independent. If the omission of a discourse connective leads to a different interpretation of the discourse relation, this means that the explicit and implicit discourse relations bound by this discourse connective are not equivalent, and the explicit discourse relation instance cannot be used to help infer the sense of the implicit discourse relation. Conversely, if the contexts for the discourse connective in explicit and implicit discourse relations do not significantly differ, then the explicit discourse relation instance can be used as weakly labeled data.

To capture this intuition, we must quantify the context differential of explicit and implicit discourse relations for each discourse connective. We represent the semantic context of a discourse

connective through a unigram distribution over words in its two arguments, with Arg1 and Arg2 combined. We use Jensen-Shannon Divergence (JSD) as a metric for measuring the difference between the contexts of a discourse connective in implicit and explicit discourse relations. Computing a context differential of the discourse connective therefore involves fitting a unigram distribution from all implicit discourse relations bound by that discourse connective and fitting another from all explicit discourse relations bound by the same discourse connective. We choose this method because it has been shown to be exceptionally effective in capturing similarities of discourse connectives (Hutchinson, 2005) and statistical language analysis in general (Lee, 2001; Ljubesic et al., 2008).

The Jensen-Shannon Divergence (JSD) metric for difference between P_o, the semantic environments (unigram distribution of words in Arg1 and Arg2 combined) in implicit discourse relations, and P_r, the semantic environments in explicit discourse relations, is defined as:

$$JSD(P_o||P_r) = \frac{1}{2}D(P_o||M) + \frac{1}{2}D(P_r||M)$$

where $M = \frac{1}{2}(P_o+P_r)$ is a mixture of the two distributions and $D(.||.)$ is Kullback-Leibler divergence function for discrete probability distributions:

$$D(P||Q) = \sum_i \ln(\frac{P(i)}{Q(i)})P(i)$$

4.1.4 Discourse Connective Classification

Using the two metrics, we can classify discourse connectives into the following classes:

1. Freely omissible: High OR and low JSD
2. Omissible: Low non-zero OR and low JSD.
3. Alternating I: High OR and high JSD.
4. Alternating II: Low non-zero OR and high JSD.

5. Non-omissible: Zero OR. JSD cannot be computed because the connectives are never found in any implicit discourse relations.

Classifying the connectives into these classes allow us to empirically investigate which explicit discourse relations are useful as supplemental training data for determining the sense of implicit discourse relations. We discuss each type of connectives below.

Freely omissible discourse connectives

These are connectives whose usage in implicit and explicit discourse relations is indistinguishable and therefore suitable as a source of supplemental training data. These connectives are defined as having high omission rate and low context differential. This definition implies that the omission is frequent and insensitive to the context. "Because" and "in particular" in (9) and (10) are such connectives. Dropping them has minimal impact on the understanding the discourse relation between their two arguments and one might argue they even make the sentences sound more natural.

(9) We cleared up questions and inconsistencies very quickly because the people who had the skills and perspective required to resolve them were part of the task team. (WSJ0562)

(10) Both companies are conservative marketers that rely on extensive market research. P&G, in particular, rarely rolls out a product nationally before extensive test-marketing. (WSJ0589)

Omissible discourse connectives

They are connectives whose usage in implicit and explicit discourse relations is indistinguishable, yet they are not often omitted because the discourse relation might be hard to interpret without them. These connectives are defined as having low omission rate and low context differential. For example,

(11) Such problems will require considerable skill to resolve. However, neither Mr. Baum nor Mr. Harper has much international experience. (WSJ0109)

One can infer from the discourse that the problems require international experience, but Mr. Baum and Mr. Harper don't have that experience even without the discourse connective "however". In other words, the truth value of this proposition is not affected by the presence or absence of this discourse connective. The sentence might sound a bit less natural, and the discourse relation seems a bit more difficult to infer if "however" is omitted.

Alternating discourse connectives

They are connectives whose usage in implicit and explicit discourse relations is substantially different and they are defined as having high context differential. Having high context differential means that the two arguments of an explicit discourse connective differ substantially from those of an implicit discourse. An example of such discourse connectives is "nevertheless" in (12). If the discourse connective is dropped, one might infer EXPANSION or CONTINGENCY relation instead of COMPARISON indicated by the connective.

(12) Plant Genetic's success in creating genetically engineered male steriles doesn't automatically mean it would be simple to create hybrids in all crops. Nevertheless, he said, he is negotiating with Plant Genetic to acquire the technology to try breeding hybrid cotton. (WSJ0209)

We hypothesize that this type of explicit discourse relations would not be useful as extra training instances for inferring implicit discourse relations because they will only add noise to the training set.

Non-omissible discourse connectives

They are defined as discourse connectives whose omission rate is close to zero as they are never found in implicit discourse relations. For example, conditionals can not be easily expressed without the use of an explicit discourse connective like "if". We hypothesize that instances of explicit discourse relations with such discourse connectives would not be useful as additional training data

for inferring implicit discourse relations because they represent discourse relation senses that do not exist in the implicit discourse relations.

4.2 Experiments

4.2.1 Partitioning the discourse connectives

We only include the discourse connectives that appear in both explicit and implicit discourse connectives in the PDTB to make the comparison and classification possible. As a result, we only analyze 69 out of 134 connectives for the purpose of classification. We also leave out 15 connectives whose most frequent sense acccounts for less than 90% of their instances. For example, *since* can indicate a TEMPORAL sense or a CONTINGENCY sense of almost equal chance, so it is not readily useful for gathering weakly labeled data. Ultimately, we have 54 connectives as our candidates for freely omissible discourse connectives.

We first classify the discourse connectives based on their omission rates and context differentials as discussed in the previous section and partition all of the explicit discourse connective instances based on this classification. The distributions of omission rates and context differentials show substantial amount of variation among different connectives. Many connectives are rarely omitted and naturally form its own class of non-omissible discourse connectives (Figure 4.1). We run the agglomerative hierchical clustering algorithm using Euclidean distance on the rest of the connectives to divide them into two groups: high omission and low omission rates. The boundary between the two groups is around 0.65.

The distribution of discourse connectives with respect to the context differential suggests two distinct groups across the two corpora (Figure 4.2). The analysis only includes connectives that are omitted at least twenty times in the PDTB corpus, so that JSD can be computed. The hierarchical clustering algorithm divides the connectives into two groups with the boundary at around 0.32, as should be apparent from the histogram. The JSD's computed from the explicit discourse relations

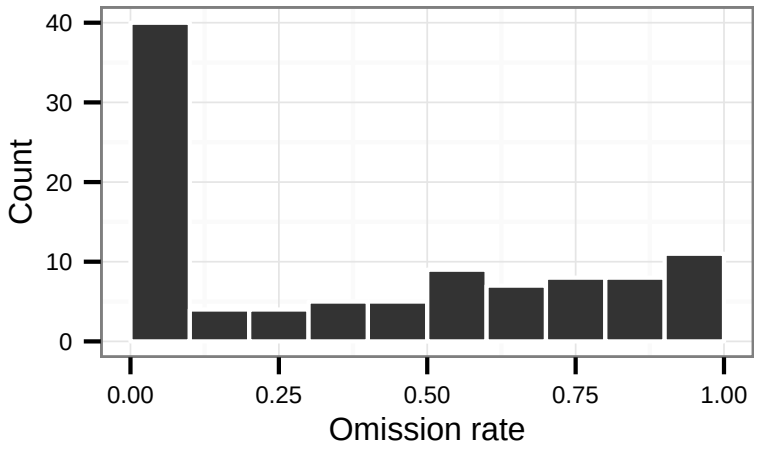

Figure 4.1: Omission rates of the discourse connective types vary drastically, suggesting that connectives vary in their optionality. Some connectives are never omitted.

from the two corpora are highly correlated ($\rho = 0.80$, $p < 0.05$), so we can safely use the Gigaword corpus for the analysis and evaluation.

The omission rate boundary and context differential boundary together classify the discourse connectives into four classes in addition to the non-omissible connectives. When plotted against each other, omission rates and context differential together group the discourse connectives nicely into clusters (Figure 4.3). For the purpose of evaluation, we combine Alternating I and II into one class because each individual class is too sparse on its own. The complete discourse connective classification result is displayed in Table 4.1.

4.2.2 Evaluation results

We formulate the implicit relation classification task as a 4-way classification task in a departure from previous practice where the task is usually set up as four *one vs other* binary classification tasks so that the effect of adding the distant supervision from the weakly labeled data can be more easily studied. We also believe this setup is more natural in realistic settings. Each classification instance consists of the two arguments of an implicit discourse relation, typically adjacent pairs of sentences in a text. The distribution of the sense labels is shown in Table 4.2. We follow the data

Class Name	OR	JSD	Connectives
Alternating I	High	High	further, in sum, in the end, overall, similarly, whereas
Alternating II	Low	High	earlier, in turn, nevertheless, on the other hand, ultimately
Freely Omissible	High	Low	accordingly, as a result, because, by comparison, by contrast, consequently, for example, for instance, furthermore, in fact, in other words, in particular, in short, indeed, previously, rather, so, specifically, therefore,
Omissible	Low	Low	also, although, and, as, but, however, in addition, instead, meanwhile, moreover, rather, since, then, thus, while
Non-omissible	zero	NA	as long as, if, nor, now that, once, otherwise, unless, until

Table 4.1: Classification of discourse connectives based on omission rate (OR) and Jensen-Shannon Divergence context differential (JSD).

Sense	Train	Dev	Test
Comparison	1855	189	145
Contingency	3235	281	273
Expansion	6673	638	538
Temporal	582	48	55
Total	12345	1156	1011

Table 4.2: The distribution of senses of implicit discourse relations in the PDTB

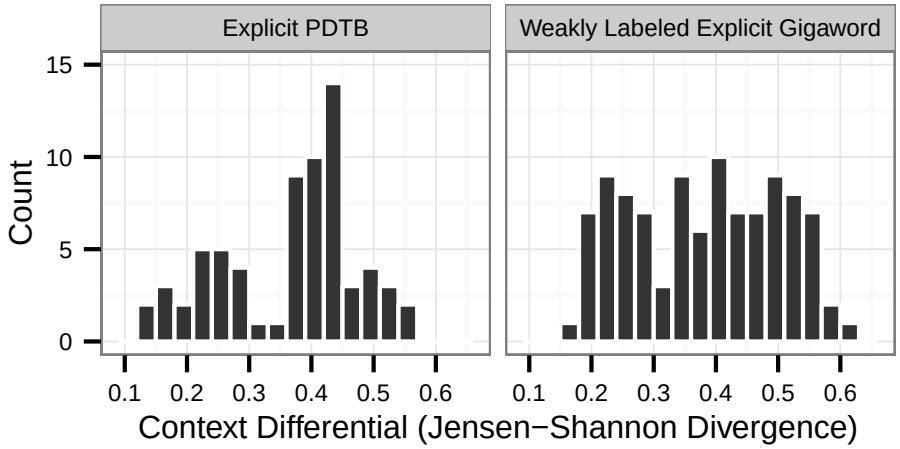

Figure 4.2: The distributions of Jensen-Shannon Divergence from both corpora shows two potential distinct clusters of discourse connectives.

split used in previous work for a consistent comparison (Rutherford and Xue, 2014). The PDTB corpus is split into a training set, development set, and test set. Sections 2 to 20 are used to train classifiers. Sections 0 and 1 are used for developing feature sets and tuning models. Section 21 and 22 are used for testing the systems.

To evaluate our method for selecting explicit discourse relation instances, we extract weakly labeled discourse relations from the Gigaword corpus for each class of discourse connective such that the discourse connectives are equally represented within the class. We train and test Maximum Entropy classifiers by adding varying number (1000, 2000, . . . , 20000) of randomly selected explicit discourse discourse relation instances to the manually annotated implicit discourse relations in the PDTB as training data. We do this for each class of discourse connectives as presented in Table 4.1. We perform 30 trials of this experiment and compute average accuracy rates to smooth out the variation from random shuffling of the weakly labeled data.

The statistical models used in this study are from the MALLET implementation with its default setting (McCallum, 2002). Features used in all experiments are taken from the state-of-the-art implicit discourse relation classification system (Rutherford and Xue, 2014). The feature set consists of combinations of various lexical features, production rules, and Brown cluster pairs. These features

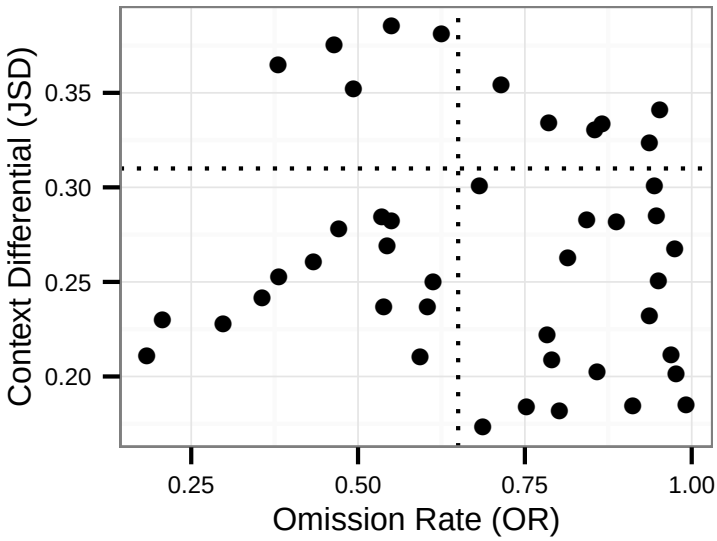

Figure 4.3: The scattergram of the discourse connectives suggest three distinct classes. Each dot represents a discourse connective.

are described in greater detail by Pitler et al. (2009) and Rutherford and Xue (2014).

Instance reweighting is required when using weakly labeled data because the training set no longer represents the natural distribution of the labels. We reweight each instance such that the sums of the weights of all the instances of the same label are equal. More precisely, if an instance i is from class j, then the weight for the instance w_{ij} is equal to the inverse proportion of class j:

$$\begin{aligned} w_{ij} &= \frac{\text{Number of total instances}}{\text{Size of class } j \cdot \text{Number of classes}} \\ &= \frac{\sum_{j'}^{k} c_{j'}}{c_j \cdot k} = \frac{n}{c_j \cdot k} \end{aligned}$$

where c_j is the total number of instances from class j and k is the number of classes in the dataset of size n. It is trivial to show that the sum of the weights for all instances from class j is exactly $\frac{n}{k}$ for all classes.

The impact of different classes of weakly labeled explicit discourse connective relations is illustrated in Figure 4.4. The results show that expicit discourse relations with freely omissible

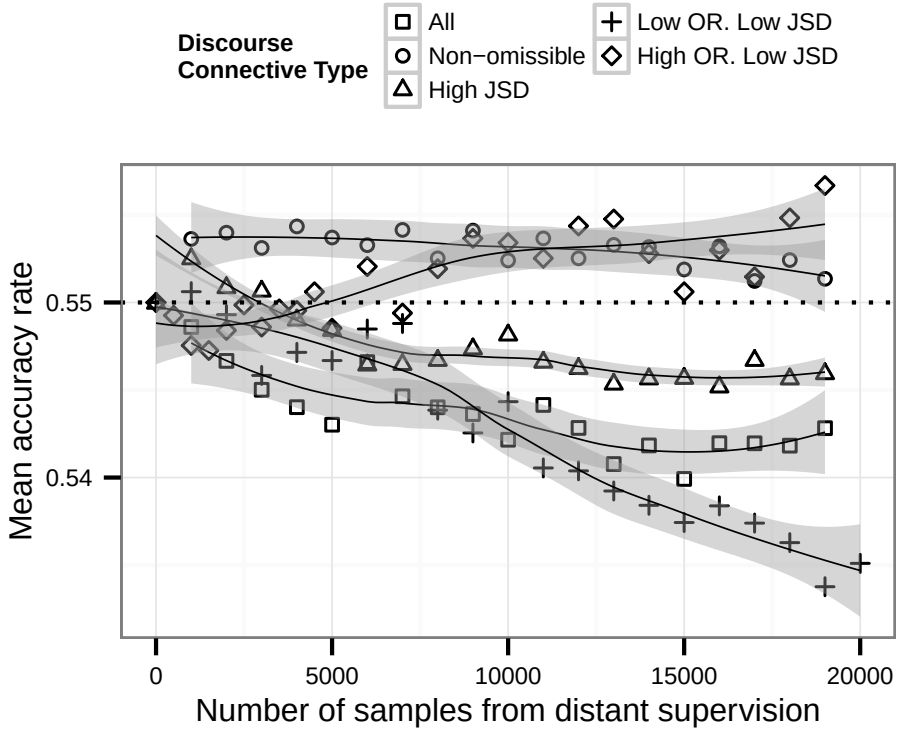

Figure 4.4: Discourse connectives with high omission rates and low context differentials lead to highest performance boost over the state-of-the-art baseline (dotted line). Each point is an average over multiple trials. The solid lines are LOESS smoothing curves.

discourse connectives (high OR and low JSD) improve the performance on the standard test set and outperform the other classes of discourse connectives and the naive approach where all of the discourse connectives are used. In addition, it shows that on average, the system with weakly labeled data from freely omissible discourse connectives continues to rise as we increase the number of samples unlike the other classes of discourse connectives, which show the opposite trend. This suggests that discourse connectives must have both high omission rates and low context differential between implicit and explicit use of the connectives in order to be helpful to the inference of implicit discourse relations.

Table 4.3 presents results that show, overall, our best performing system, the one using distant supervision from freely omissible discourse connectives, raises the accuracy rate from 0.550 to 0.571

		Baseline features	Baseline + extra data
Expansion	Precision	0.608	0.614
	Recall	0.751	0.788
	F_1	0.672	0.691
Comparison	Precision	0.398	0.449
	Recall	0.228	0.276
	F_1	0.290	0.342
Contingency	Precision	0.465	0.493
	Recall	0.418	0.396
	F_1	0.440	0.439
Temporal	Precision	0.263	0.385
	Recall	0.091	0.091
	F_1	0.135	0.147
Accuracy		0.550	**0.571**
Macro-Average F_1		0.384	**0.405**

Table 4.3: Our current 4-way classification system outperforms the baseline overall. The difference in accuracy is statistically significant ($p < 0.05$; bootstrap test).

($p < 0.05$; bootstrap test) and the macro-average F_1 score from 0.384 to 0.405. We achieve such performance after we tune the subset of weakly labeled data to maximize the performance on the development set. Our distant supervision approach improves the performance by adding more weakly labeled data and no additional features.

For a more direct comparison with previous results, we also replicated the state-of-the-art system described in Rutherford and Xue (2014), who follows the practice of the first work on this topic (Pitler et al., 2009) in setting up the task as four binary one vs. other classifiers. The results are presented in Table 4.4. The results show that the extra data extracted from the Gigaword Corpus is particularly helpful for minority classes such as *Comparison vs. Others* and *Temporal vs Others*, where our current system significantly outperforms that of Rutherford and Xue (2014). Interestingly, the *Expansion vs. Others* classifier did not improve as the *Expansion* class in the four-way classification (Table 4.3).

	R&X (2014)	Baseline + extra data	Baseline
Comparison vs Others	0.397	**0.410**	0.380
Contingency vs Others	0.544	0.538	0.539
Expansion vs Others	0.702	0.694	0.679
Temporal vs Others	0.287	**0.333**	0.246

Table 4.4: The performance of our approach on the binary classification task formulation.

Class	Gigaword only	Gigaword + Implicit PDTB
Freely omissible	**0.505**	**0.571**
Omissible	0.313	0.527
Alternating I + II	0.399	0.546
Non-Omissible	0.449	0.554
All of above	0.490	0.547

Table 4.5: The accuracy rates for the freely omissible class are higher than the ones for the other classes both when using the Gigaword data alone and when using it in conjunction with the implicit relations in the PDTB.

4.2.3 Just how good is the weakly labeled data?

We performed additional experiments to get a sense of just how good the weakly labeled data extracted from an unlabeled corpus are. Table 4.5 presents four-way classification results using just the weakly labeled data from the Gigaword Corpus. The results show that the same trend holds when the implicit relations from the PDTB are not included in the training process. The freely omissible discourse connectives achieves the accuracy rate of 0.505, which is significantly higher than the other classes, but they are weaker than the manually labeled data, which achieves the accuracy rate of 0.550 for the same number of training instances.

Weakly labeled data are not perfectly equivalent to the true implicit discourse relations, but they do provide strong enough additional signal. Figure 4.5 presents experimental results that compare the impact of weakly labeled data from Gigaword Corpus vs gold standard data from the PDTB for the freely omissible class. The mean accuracy rates from the PDTB data are

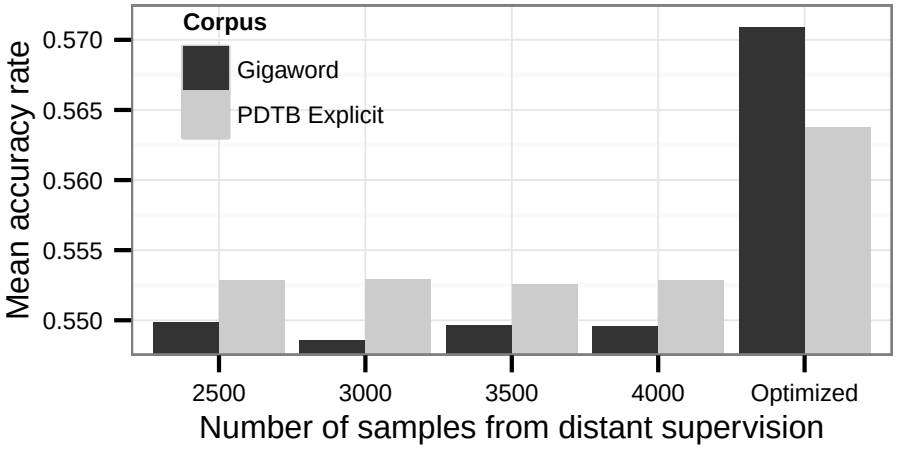

Figure 4.5: The PDTB corpus leads to more improvement for the same amount of the data. However, Gigaword corpus achieves significantly better performance ($p < 0.05$; bootstrap test) when both models are tuned on the developement set.

significantly higher than those from the Gigaword Corpus (p <0.05; t-test and bootstrap test) for the same number of training instances combined with the implicit discourse relations. However, when the number of introduced weakly labeled data exceeds a certain threshold of around 12,000 instances, the performance of the Gigaword corpus rises significantly above the baseline and the explicit PDTB (Figure 4.4).

The relative superiority of our approach derives precisely from the two selection criteria that we propose. The performance gain does not come from the fact that freely omissible discourse connectives have better coverage of all four senses (Table 4.6). When all classes are combined equally, the system performs worse as we add more samples although all four senses are covered. The coverage of all four senses is not sufficient for a class of discourse connectives to boost the performance. The two selection criteria are both necessary for the success of this paradigm.

4.3 Related work

Previous work on implicit discourse relation classification have focused on supervised learning approaches (Lin et al., 2010a; Rutherford and Xue, 2014), and the distantly supervised approach

	Sense			
Class	Comp.	Cont.	Exp.	Temp.
Freely omissible	2	6	10	1
Omissible	4	2	5	3
Alternating I	1	0	5	0
Alternating II	2	0	0	3
Non-omissible	0	3	3	2

Table 4.6: The sense distribution by connective class.

using explicit discourse relations has not shown satisfactory results (Pitler et al., 2009; Park and Cardie, 2012; Wang et al., 2012; Sporleder and Lascarides, 2008) Explicit discourse relations have been used to remedy the sparsity problem or gain extra features with limited success (Biran and McKeown, 2013; Pitler et al., 2009). Our heuristics for extracting discourse relations has been explored in the unsupervised setting (Marcu and Echihabi, 2002b), but it has never been evaluated on the gold standard data to show its true efficacy. Our distant supervision approach chooses only certain types of discourse connectives to extract weakly labeled data and is the first of its kind to improve the performance in this task tested on the manually annotated data.

Distant supervision approaches have recently been explored in the context of natural language processing due to the recent capability to process large amount of data. These approaches are known to be particularly useful for relation extraction tasks because training data provided do not suffice for the task and are difficult to obtain (Riloff et al., 1999; Yao et al., 2010). For example, Mintz et al. (2009) acquire a large amount of weakly labeled data based on the Freebase knowledge base and improves the performance of relation extraction. Distantly supervised learning has also recently been demonstrated to be useful for text classification problems (Speriosu et al., 2011; Marchetti-Bowick and Chambers, 2012). For example, Thamrongrattanarit et al. (2013) use simple heuristics to gather weakly labeled data to perform text classification with no manually annotated training data.

Discourse connectives have been studied and classified based on their syntactic properties such subordinating conjunction, adverbials, etc. (Fraser, 2006; Fraser, 1996). While providing a useful

insight into how discourse connectives fit into utterances, the syntactic classification does not seem suitable for selecting useful discourse connectives for our purposes of distant supervision for our task.

4.4 Conclusion and Future Directions

We propose two selection criteria for discourse connectives that can be used to gather weakly labeled data for implicit discourse relation classifiers and improve the performance of the state-of-the-art system without further feature engineering. As part of this goal, we classify discourse connectives based on their distributional semantic properties and found that certain classes of discourse connectives cannot be omitted in every context, which plague the weakly labeled data used in previous studies. Our discourse connective classification allows for the better selection of data points for distant supervision.

More importantly, this work presents a new direction in distantly supervised learning paradigm for implicit discourse relation classification. This virtual dramatic increase in the training set size allows for more feature engineering and more sophisticated models. Implicit discourse relation classification is now no longer limited to strictly supervised learning approaches.

Chapter 5

Neural Discourse Mixture Model

Sentences in the same body of text must be coherent. To compose a paragraph that "makes sense," the sentences must form a discourse relation, which manifests the coherence between the sentences. Adjacent sentences can form a discourse relation even with the absence of discourse connectives such as *because*, which signals a causal relation, or *afterwards*, which signals a temporal relation. These implicit discourse relations abound in naturally occurring text, and people can infer them without much effort. Although crucial for summarization systems (Louis et al., 2010), and text quality assessment (Pitler and Nenkova, 2008; lou, 2014) TODO, among many other applications, the task of classifying implicit discourse relations remains a challenge for an automatic discourse analysis.

The main difficulty of modeling and classifying discourse relations lies in the problem of representation, which feature engineering might not be able to fully address. Traditional approaches to discourse relation classification involve loading a Naive Bayes classifier with a battery of features extracted from syntactic parses and various lexicons (Pitler et al., 2009; Park and Cardie, 2012). The best performing system in this paradigm replaces all the words with their Brown cluster assignment (Brown et al., 1992) and uses their cartesian product as features (Rutherford and Xue, 2014). This method is effective because it reduces the sparsity in the feature space while introduc-

ing latent semantic features. Sparsity reduction is one of the main motivations for our work and previous works in this domain (Biran and McKeown, 2013; Li and Nenkova, 2014).

Distributed representation has been successfully employed to enhance many natural language systems due to new modeling techniques and larger computational capacity. This representation treats a word or a feature as a vector of real-valued numbers which are derived from massive amount of unannotated data. No longer treated as a discrete atomic symbol, a word or a feature has more expressive power, which improves the performance of many applications. Notably, word-level distributed representation has been shown to improve fact extraction (Paşca et al., 2006), query expansion (Jones et al., 2006), and automatic annotation of text (Ratinov et al., 2011). We would like to investigate the effectiveness of distributed representation in modeling discourse relations. In this paper, we propose a distributed discourse model composed from individual word vectors and aim to use the model to classify implicit discourse relations.

We use a neural network model architecture to combine the computational efficiency of Naive Bayes models and the expressiveness of word vectors. Naive Bayes is the most efficient and consistently best-performing model thus far for discourse relation classification, despite the use of millions of features (Pitler et al., 2009; Park and Cardie, 2012; Rutherford and Xue, 2014). Multilayer neural networks, on the other hand, cannot handle a large number of features without enormous computational resources, but their structure allows for the capability to take advantage of word vectors and intermediate representations of a discourse vector, which might benefit classification.

Our contributions from this work are three-fold.

1. We propose the Neural Discourse Mixture Model as the new state-of-the-art implicit discourse relation classification and the first neural discourse model for written text. The model improves the performance of the implicit discourse relation classifier by a significant margin compared to Rutherford and Xue (2014) baseline.

2. Through series of experiments, we establish that neural word embeddings and their intermediate abstraction provide better representation of discourse relations than Brown cluster

pairs alone. This suggests that neural network model is a promising direction for discourse analysis and classification.

3. We propose and use Skip-gram Discourse Vector Model to model each discourse relation as a discourse vector composed from individual Skip-gram vectors. Cluster analysis of the model reveals many linguistic phenomena implicated in discourse relations.

5.1 Corpus and Task Formulation

The Penn Discourse Treebank (PDTB) is a layer of annotation on top of Wall Street Journal articles in the Penn Treebank (Prasad et al., 2008; Marcus et al., 1993). A discourse relation is defined as a local coherence relation between the two arguments, which can be a sentence, a constituent, or an arbitrary incontiguous span of text. Implicit discourse relations are relations whose two arguments (called Arg1 and Arg2) are coherent without the use of discourse connectives such as *although*, *and*, and *when*. The corpus contains roughly the same number of implicit and explicit discourse relations.

The types or senses of discourse annotations are also annotated. The senses more specifically describe the way in which Arg1 and Arg2 are related. The sense inventory is organized hierarchically. The top level senses are Comparison, Contingency, Expansion, and Temporal. Our work focuses on the top level senses because they are the four discourse relations that various discourse analytic theories agree on (Mann and Thompson, 1988) and they contain relatively larger numbers of tokens for training and evaluating. Consistent with the previous work, the task is formulated as four separate one-vs-all classification problems. We use the same model architecture for all four binary classification tasks.

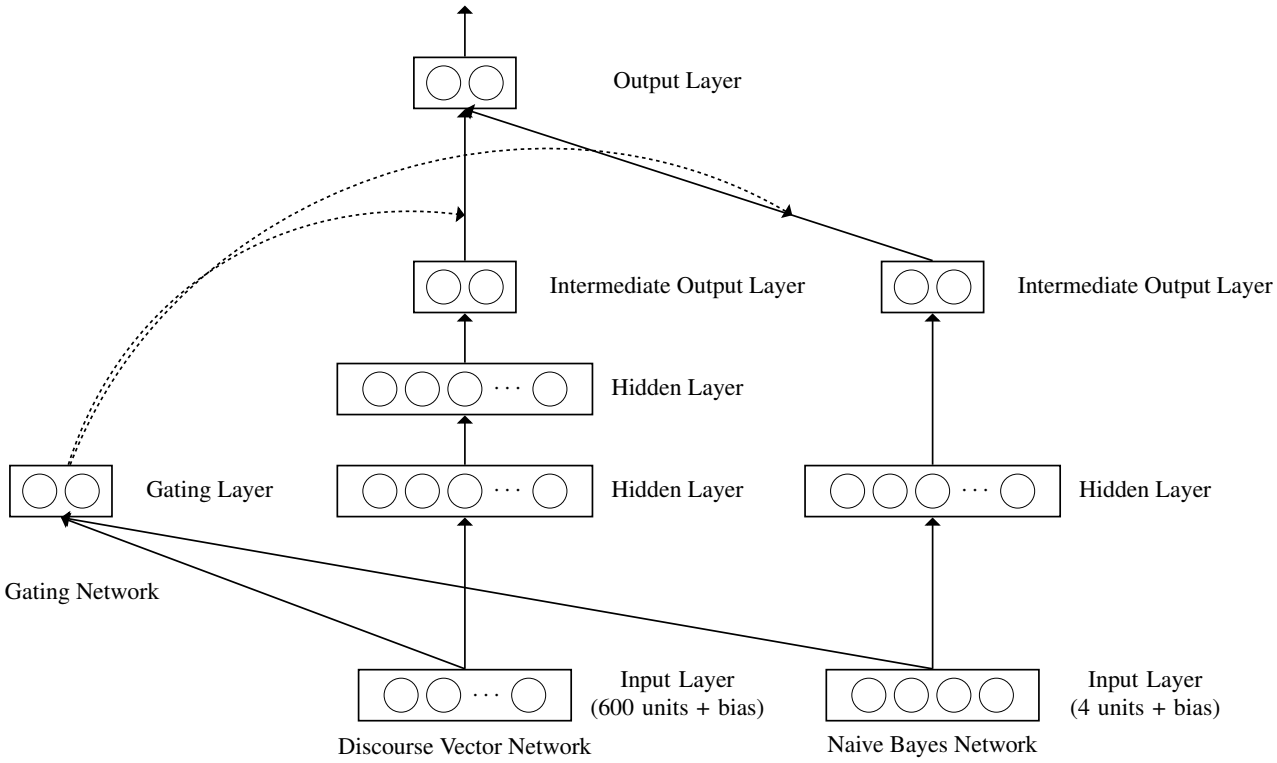

Figure 5.1: Schematic depiction of the architecture of the Neural Discourse Mixture Model

5.2 The Neural Discourse Mixture Model Architecture

The Neural Discourse Mixture Model (NDMM) makes use of the strength of semantic abstraction achieved by distributed word vector representation and combines it with the binary features traditionally used in Naive Bayes implicit discourse relation classification. The main motivation of the NDMM derives from exploiting multiple levels of representations: discrete-valued Brown word cluster pairs and continuous-valued Skip-gram word embeddings. Brown word cluster pairs are shown to capture many high-level discursive phenomena but suffer from the sparsity problem (Rutherford and Xue, 2014). On the other hand, distributed word vectors are not sparse and lend themselves to semantic abstraction through neural network models. When used together, they should provide us with extra information that captures the characteristics of each type of discourse relation.

The architecture of the NDMM takes advantage of multi-level representation by integrating

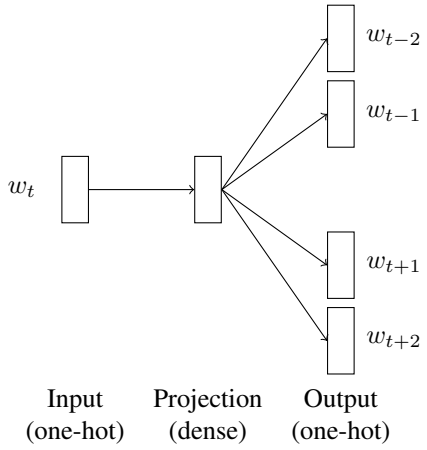

Figure 5.2: In the Skip-gram architecture, word vectors (projection) are induced such that one can use the learned weights to predict the co-occurring words (output).

a component from the Mixture of Experts model (Jacobs et al., 1991). The model consists of two standard multilayer neural networks and a gating network that mixes the output from the two models (Figure 5.1). The number of hidden layers have been optimized and tuned on the development set to attain the best performance. These three main components of NDMM are detailed in the following subsections.

5.2.1 Discourse Vector Network

The idea behind the discourse vector network is that we want to build an abstract representation of a discourse relation in a bottom-up manner from word to discourse argument and to discourse vectors. A discourse argument (Arg1 or Arg2) vector is formed by adding up Skip-gram word vectors. A discourse vector – the input to the network – is the concatenation of Arg1 and Arg2 vectors. An abstract representation vector is formed in the hidden layers by a linear combination of the elements in the discourse vector.

The Skip-gram model is an efficient method for learning high-quality distributed word vector representation (Mikolov et al., 2013a; Mikolov et al., 2013b). A Skip-gram word vector is induced such that the values in the vector encode co-occurring words within a certain window (Figure

5.2). One of the attractive properties of Skip-gram word vectors is additive compositionality. For example, the vector composed by adding up the vectors for "Vietnam" and "capital" is closest to the vector for "Hanoi" among the other vectors in the vocabulary. We exploit this property to construct the input layer.

The input layer for this model is the concatenation of two vectors, which represent Arg1 and Arg2 in the discourse relation in question. Each word in the vocabulary is associated with a word vector learned by a Skip-gram model on a separate large unannotated corpus. Each argument vector is composed by adding all of the word vectors in the argument element-wise. These two argument vectors A_1 and A_2 are then concatenated to form a discourse vector. More precisely, if Arg i has n words $w_{i1}, w_{i2}, \ldots w_{in}$ and $V_x \in \mathbb{R}^d$ is a word vector for word x learned by a Skip-gram model, then

$$A_i = \sum_{t=1}^{n} V_{w_{it}} .$$

A Skip-gram discourse vector $X_s \in \mathbb{R}^{2d}$ is the concatenation of A_1 and A_2:

$$X_s = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$$

The network has two hidden layers. The first hidden layer is a standard non-linearly activated hidden layer connected to the input layer filled with a discourse vector:

$$H_1 = \sigma(X_s \cdot W_1^s)$$

where $H_1 \in \mathbb{R}^k$ is a hidden layer with k units, $W_1 \in \mathbb{R}^{2d \times k}$ is the associated weight matrix, and $\sigma : \mathbb{R} \rightarrow [0, 1]$ is a non-linear activation function such as rectified linear unit activation function or sigmoid function.

Similarly, the second hidden layer is a standard non-linearly activated hidden layer of the same

size and feeds forward to the softmax-activated output layer:

$$H_2 = \sigma(H_1 \cdot W_2^s)$$
$$Y_s = \text{softmax}(H_2 \cdot W_3^s)$$

where $H_2 \in \mathbb{R}^k$ is the second hidden layer, and $W_2 \in \mathbb{R}^{k \times k}$ is the associated weight matrix. $W_3 \in \mathbb{R}^{k \times 2}$ is the weight matrix for the softmax output layer, which outputs the probability vector $Y_s \in \mathbb{R}^2$ over the two binary labels.

5.2.2 Naive Bayes Network

We enhance the previous approaches by further processing the outputs of the four binary Naive Bayes classifiers through a hidden layer of a neural network model. One motivation to use a Naive Bayes Network with a hidden layer is to potentially downweight the influence of the confusable labels. For example, we note that EXPANSION and TEMPORAL labels are easily confusable. Depending on the training process, the hidden layer might act like an XOR function over these two labels and downweight the probability output of the Naive Bayes Network.

Another motivation for a Naive Bayes Network is to cut the computational cost incurred by the large feature set. Unlike Naive Bayes classifiers, multilayer neural network models cannot support the large feature set required by the previous work because the number of parameters becomes prohibitively large compared to the dataset size when connected to a hidden layer. Instead of directly injecting this gigantic feature set into the model, we train a set of one-vs-all Naive Bayes classifiers and feed their outputs to a neural network for further processing.

Four one-vs-all Naive Bayes classifiers are trained. We use all of the features from Rutherford and Xue (2014). The combined outputs $X_b \in \{0, 1\}^4$ from the four classifiers are used as the input

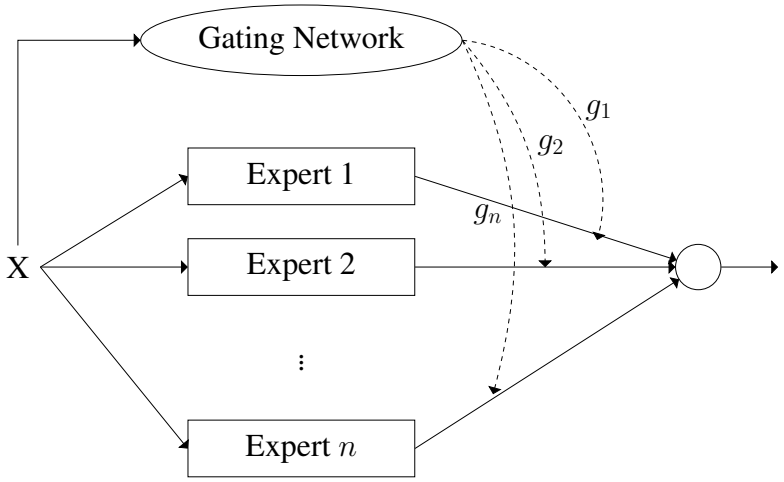

Figure 5.3: The general model architecture of Mixture of Experts model. The outputs from experts 1 through n are aggregated based on the mixing proportions $g_1, g_2, \ldots, g_n$, which are calculated from the pattern of the input X.

to the network:

$$X_b = \begin{bmatrix} B_{expansion} \\ B_{comparison} \\ B_{contingency} \\ B_{temporal} \end{bmatrix},$$

where B_l is the output from the Naive Bayes classifier for label l against the other labels. $B_l = 1$ if the classifier for label l classifies that discourse relation as l. $B_l = 0$ otherwise.

The network has one hidden layer and one output layer. The hidden layer contains 10 units. The output layer yields a probability vector Y_b over the positive and negative label through softmax function:

$$Y_b = \text{softmax}(\sigma(X_b \cdot W_1^b) \cdot W_2^b),$$

where $W_1^b \in \mathbb{R}^{4\times10}$ and $W_2^b \in \mathbb{R}^{10\times2}$ are associated weight matrices.

5.2.3 Gating Network

The gating network acts to optimally allocate weights among the sub-modular networks based on the input vector. The gating network effectively selects the appropriate training samples for each of

the submodules during training and selects the outputs from the appropriate sub-modules during classification. We hypothesize that a Naive Bayes Network might perform better than a discourse vector network when Arg1 and Arg2 are long, which might deteriorate the compositionality of Skip-gram vectors. This gating or reweighting process and the training process of the two sub-modular networks are data-driven and done simultaneously through the Mixture of Experts architecture (Figure 5.3)

The Mixture of Experts model is a neural network model that consists of multiple sub-modular "expert" neural networks and a gating network that assigns weights for each expert network based on the input (Jacobs et al., 1991). During training, the gating network learns to detect which subset of input pattern is suitable for which expert and simultaneously makes each expert become specialized in a certain subset of input patterns. Therefore, the parameters in a sub-modular expert network will turn out differently when trained in a Mixture of Experts model than when trained on its own. During classification, the gating network decides which expert should receive more vote and then combines the classification results from the experts accordingly.

The NDMM can be thought of as an instance of the Mixture of Experts model. The two sub-modular networks are the discourse vector network and the Naive Bayes Network as described in the previous sections. The input for the gating network is the concatenation of Skip-gram discourse vector X_s and the input vector for the Naive Bayes Network X_b defined above. The gating layer output Y_g is softmax-activated:

$$Y_g = \sigma(\begin{bmatrix} X_s \\ X_b \end{bmatrix} \cdot W^g).$$

Each value of Y_g corresponds to the weight that a sub-modular network receives. The final output Y is a weighted average of the output from the two sub-modular networks:

$$Z = [Y_s, Y_b]$$

$$Y = Z \cdot Y_g,$$

where $Z \in \mathbb{R}^{2\times 2}$ and Y_s, Y_b, and $Y_g \in \mathbb{R}^2$ are column vectors as described previously.

5.2.4 Training Procedure

The gating network, discourse vector network, and Naive Bayes network are trained simultaneously within the same network using the backpropagation algorithm. The objective function is a weighted cross-entropy loss function. Word vectors used in composing discourse vectors are trained separately on a separate corpus and fixed during the training process.

5.3 Experimental Setup

The experimental setup is consistent with the state-of-the-art baseline system in classifying implicit discourse relations (Rutherford and Xue, 2014). The PDTB corpus is split into training set, development set, and test set accordingly. Sections 2 to 20 are used for training the networks (12,345 instances). Sections 0–1 are used for developing feature sets and tuning hyper parameters (1,156 instances). Section 21–22 are used for testing the systems (1,011 instances).

Each instance of an implicit discourse relation consists of the text spans from the two arguments. The task is formulated as one-vs-all classification task. The label is either positive or negative, where the positive label is the label of interest such as TEMPORAL relation and the negative label is NON-TEMPORAL.

We follow the recipe set out by Rutherford and Xue (2014) to create a baseline Naive Bayes classifier. All of the linguistic pre-processing tasks are done automatically to generate some of the features that Naive Bayes classifiers require. We use the Stanford CoreNLP suite to lemmatize and part-of-speech tag each word (Toutanova et al., 2003; Toutanova and Manning, 2000), obtain the phrase structure and dependency parses for each sentence (De Marneffe et al., 2006; Klein and Manning, 2003), identify all named entities (Finkel et al., 2005), and resolve coreference (Raghunathan et al., 2010; Lee et al., 2011; Lee et al., 2013). We use MALLET implementation of Naive

Bayes classifier (McCallum, 2002). The features that this Naive Bayes baseline and the Naive Bayes Network have in common are taken from the project site to generate exactly identical feature sets.

The Skip-gram word vector dictionary in this experiment is induced from a Google News dataset of about 100 billion words[1]. The dictionary contains 300-dimensional vectors for 3 million words and phrases. Words that are not in the dictionary are excluded and ignored when composing discourse vectors.

Mini-batch backpropagation algorithm and AdaGrad algorithm were used to train the model (Duchi et al., 2011b). We tune all of the hyperparameters on the development set: the activation functions (sigmoid or rectified linear unit), number hidden layers (1, 2, 3 layers), number of hidden units (100, 500, 700, 1,000, 1,200, 1,500 units), meta-learning rate for AdaGrad (0.001, 0.05, 0.01), and mini-batch size (5, 10, 20, 30). We train the model with and without unsupervised pre-training (Hinton et al., 2006).

The algorithms are implemented in the high-level dynamic programming language Julia (Bezanson et al., 2012). The code, along with the configuration files and all of the feature matrices, is made available at `www.github.com/[anonymized]`. The implementation provided on the project site is flexible enough to generalize to other datasets.

We evaluate the NDMM against four baseline systems (detailed in the following subsections) to tease apart the comparative effectiveness of individual sub-modules in the network.

5.3.1 Naive Bayes classifier with Brown cluster pairs only (NB+B)

We want to test the efficacy of Brown cluster pair features as the representation of discourse relation. So far Brown cluster pair features are one of the more effective features for this task (Rutherford and Xue, 2014). To generate Brown cluster pair representation, we replace the words with their Brown cluster assignment induced by Brown word clustering algorithm (Brown et al., 1992; Turian et al., 2010). Brown cluster pairs are the cartesian product of the set of Brown clusters in Arg1

[1]freely available at `code.google.com/p/word2vec/`

and the set of Brown clusters from Arg2. For example, if Arg1 has two words $w_{1,1}, w_{1,2}$, and Arg2 has three words $w_{2,1}, w_{2,2}, w_{2,3}$, then the word pair features are $(b_{1,1}, b_{2,1}), (b_{1,1}, b_{2,2}), (b_{1,1}, b_{2,3})$, $(b_{1,2}, b_{2,1}), (b_{1,2}, b_{2,2}), (b_{1,2}, b_{2,3})$, where $b_{i,j}$ is Brown cluster assignment for $w_{i,j}$.

For our purpose, the implementation of feature generation is taken from the code on the project site provided by Rutherford and Xue (2014).

5.3.2 Rutherford & Xue 2014 baseline (NB+B+L)

The comparison against this strongest baseline shows us whether the NDMM works better than the simpler traditional approach or not. This baseline uses Naive Bayes classifiers loaded with Brown cluster pairs (like the baseline in 5.3.1) and other linguistic features. We hypothesize that the NDMM should outperform this – and every other – baseline because it adapts model training process to the use of multiple levels of discourse relation representation. In addition to Brown cluster pairs, the feature set consists of combinations of first, last, and first 3 words, numerical expressions, time expressions, average verb phrase length, modality, General Inquirer tags, polarity, Levin verb classes, and production rules. These features are described in greater detail in Pitler et al. (2009).

5.3.3 Naive Bayes Network baseline (NBN)

We want to compare the NDMM against this Naive Bayes network sub-module to evaluate the margin of improvement before adding the discourse vector network and the gating network. We separate out this sub-module of the NDMM and train it without the gating layer. The network architecture is described in Section 5.2.2. The input layer consists of the outputs from the four one-vs-all best-performing classifiers. The output layer of the Naive Bayes network is the final output layer that decides the final classification.

Best discourse vector network configuration when trained in the NDMM

Task	Hidden layers	Batch size
COMPARISON	2×700 units	5
CONTINGENCY	2×100 units	20
EXPANSION	2×100 units	10
TEMPORAL	2×1000 units	5

Best discourse vector network configuration when trained separately

Task	Hidden layers	Batch size
COMPARISON	2×500 units	30
CONTINGENCY	2×1000 units	5
EXPANSION	2×1200 units	20
TEMPORAL	2×1500 units	10

Table 5.1: The number of hidden units and the mini-batch sizes are tuned on the development set. When trained separately, discourse vector network requires many more hidden units to attain the best results.

5.3.4 Discourse Vector Network baseline (DVN)

We want to test the efficacy of our discourse vectors in representing discourse relations for the purpose of discourse relation classification. We hypothesize that the performance of this baseline should be better than all previous baselines because of superior representational power. To test our hypothesis, the discourse vector network is trained separately without the gating layer. The output layer of the discourse vector network makes the final classification decision.

5.4 Results and Discussion

The NDMM converges smoothly on the order of hours, on a single core. The results are robust against random initialization. Unsupervised pre-training consistently hurts the performance, so we leave out the results from the experiments with unsupervised pre-training. A possible explanation is that our hidden layers are not large enough to benefit from the pre-training routine (Erhan et al., 2010). The best performing configuration for each classification task is shown in Table 5.1. The best meta-learning rate is 0.001. Sigmoid function and Rectified Linear Unit (ReLU) function yield

System	Section	Architecture	Representation of discourse relation
NB+B	5.3.1	Naive Bayes	Brown cluster pairs
NB+B+L	5.3.2	Naive Bayes	Brown cluster pairs + other linguistics features
NBN	5.2.2	Neural Net	4-dimensional output from NB+B+L
DVN	5.2.1	Neural Net	Distributed discourse vector built up from Skip-gram word vectors
NDMM	5.2	Mixture of Experts	Mixture of NBN and DVN

System	Comparison vs others			Contingency vs others			Expansion vs others			Temporal vs others		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
NB+B	29.58	48.97	36.88	45.66	59.71	51.75	64.84	59.29	61.94	12.70	70.91	21.54
NB+B+L	27.34	72.41	39.70	44.52	69.96	54.42	59.59	85.50	70.23	18.52	63.64	28.69
NBN	31.16	49.65	38.29	47.20	68.13	55.77	58.27	90.33	70.84	19.52	60.00	29.46
DVN	32.50	63.44	42.99	46.41	71.06	56.15	63.14	65.61	64.35	26.60	52.72	**35.36**
NDMM	35.24	55.17	**43.01**	49.60	69.23	**57.79**	59.47	88.10	**71.01**	26.60	52.72	**35.36**

Table 5.2: The Neural Discourse Mixture Model improves the performance of all four classification tasks. The discourse vector network performs comparably for Comparison and Temporal relations.

similar results. Only the results from ReLU are shown.

The NDMM improves the performance on all four discourse relation types in the test set (Table 5.2). When trained separately, the discourse vector network and the Naive Bayes network perform worse than the NDMM for all tasks. The Comparison and Temporal relations receive the highest boost in performance. The NDMM successfully combines the strength of the two sub-modular networks and raises the bar for the implicit relation classification.

Distributed discourse vector representation embedded in our new approach indeed provides a superior representation for discourse relation to Brown cluster pairs. The DVN alone outperforms the Naive Bayes model with Brown cluster pair features by 7% absolute on average. As simple as it seems, discourse vectors composed from summing Skip-gram word vectors are very effective in representing discursive information within the relation.

More strikingly, the DVN in isolation performs well in comparison to the NDMM given its small number of features. The DVN uses only 600 continuous-valued features, which is minuscule compared to most NLP tasks. The network achieves slightly inferior yet comparable results for all categories except for the Expansion relations. The strength of this network lies in the hidden layers,

whose weights are learned from the data. Most of the heavy lifting is done in the hidden layers. As a consequence, the best configuration for isolated DVN uses substantially more hidden units than the DVN trained within the NDMM, probably to compensate for the information from the Naive Bayes Network (Table 5.1). These results suggest that implicit discourse relation classification does not necessarily require manual feature engineering effort to attain similar performance. If we can represent the discourse relation in a sensible or linguistically-motivated fashion, we can build a model that is less reliant on expensive manually annotated resources.

5.5 Skip-gram Discourse Vector Model

Now that we have demonstrated the capacity of the NDMM and the DVN as classification models, we would like to investigate the effectiveness of Skip-gram discourse vector space as a discourse model. Vector addition seems like a very naive way to compose individual word vectors into a sentence or a discourse as the word ordering is totally disregarded. However, it is evident from the previous experiment that this discourse representation is comparable if not better than the largely manually crafted feature vector when used in a classification setting. If a discourse vector used by the NDMM is indeed a decent representation of a discourse relation in question, then a set of discourse relations should cluster in a linguistically interpretable manner to some degree.

We conduct a clustering analysis to test whether Skip-gram discourse vectors are valid representations of discourse relations. We convert all of the implicit discourse relations in the training set into Skip-gram discourse vectors as defined in Section 5.2.1. Then we run hierarchical clustering with respect to Euclidean distance. The distance between two discourse vectors X_1 and X_2 is simply the L_2 norm of the difference between the two vectors: $||X_1 - X_2||_2$. The resulting hierarchy is cut such that 500 clusters are induced.

Close inspection of individual clusters provide some insight into how Skip-gram discourse vectors help with discourse relation classification. In general, we do not notice any clusters that share similar

syntactic structures. Most clusters seem to be organized semantically. For example, this cluster contains 27 discourse relations both of whose arguments have negative sentiment.

(13) The market also moved at early afternoon on news **that Jaguar shares were being temporarily suspended at 746 pence ($11.80) each.** Secretary of State for Trade and Industry Nicholas Ridley said later in the day *that the government would abolish its golden share in Jaguar, ...* (WSJ0231)

(14) **"We're trading with a very wary eye on Wall Street,"** said Trevor Woodland, chief corporate trader at Harris Trust & Savings Bank in New York. "*No one is willing to place a firm bet that the stock market won't take another tumultuous ride.*" (WSJ1931)

All of the examples of discourse relations follow this format: Arg1 is bold-faced, Arg2 is italicized, and the WSJ section numbers are shown in the parentheses at the end. Previous baseline systems make use of an external lexicon to compute the polarity score of the two arguments and use them as features for the Naive Bayes classifier (Pitler et al., 2009). It turns out that Skip-gram discourse vectors can cleanly capture the sentiment contrast or continuity that runs across the two arguments without elaborate lexicon.

Other dominant clustering patterns are the ones organized by topics or genres. The two arguments in the discourse relations in such clusters discuss the same topic. Some examples of such clusters are shown in Table 5.3. Generality-specificity shift has been found to be helpful features for this task and most likely to be a linguistic phenomenon found in certain types of discourse relations (Rutherford and Xue, 2014). Along the same line with the previous finding, the absence of topic shift detected by Skip-gram discourse vectors might provide some clues to the types of discourse relations.

We have shown that Skip-gram discourse vectors can capture some of the linguistic phenomena found in certain types of implicit discourse relations, but the question is left whether this property directly contributes to the improvement in the classification tasks. We hypothesize that the data

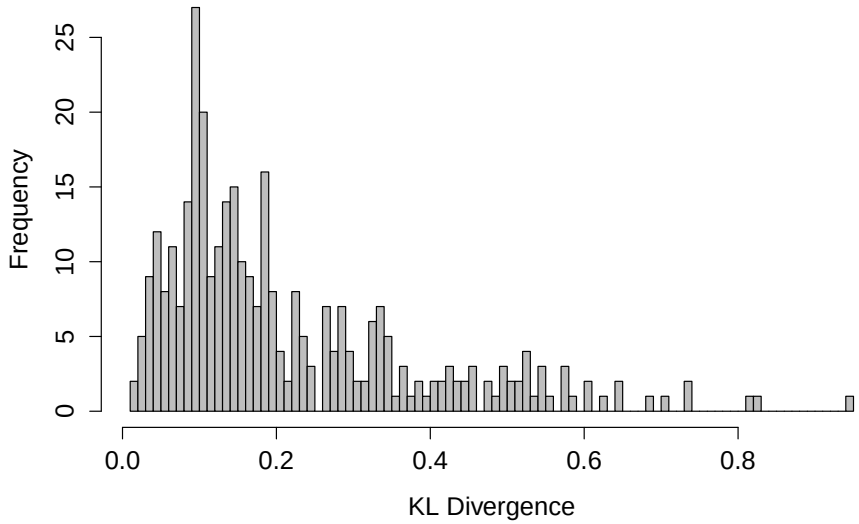

Figure 5.4: Most of the KL divergence values are reasonably far from zero. The label distributions of most of the one thousand clusters are substantially different from the marginal distribution.

are grouped or clustered in the Skip-gram discourse vector space in such a way that each cluster of discourse relations is easy to classify. If the distribution of relation types after clustering is very different from the distribution of relation types without clustering (the actual distribution of labels in the training set), then it is suggested that the Skip-gram discourse vector provides a decent representation for classification purposes.

To test this hypothesis, we run k-means clustering algorithm with respect to Euclidean distance to partition the training set into 500 clusters and then compare the distribution within each cluster with the actual distribution of the implicit discourse relation types in the dataset. If Skip-gram discourse vector representation is poor, then the distribution within the cluster should be as random as the distribution of labels without clustering. We quantify the difference between the two distributions by computing Kullback-Leibler (KL) divergence:

$$D_{KL}(P||Q) = \sum_{l} \log_2 \frac{P(l)}{Q(l)} P(l)$$

where $P(\cdot)$ is the empirical probability distribution function of the relation types in the cluster, $Q(\cdot)$ is the distribution function of the relation types of all implicit discourse relations in the training set, and l is a label. If $D_{KL}(P||Q)$ is high, then we can infer that the actual distribution and the distribution within the cluster are substantially different.

The distribution of labels within each cluster is on average significantly different from the actual distribution of labels. The mean KL divergence is 0.2131 and significantly different from zero (t-test; s.e. $\pm$ 0.0179. $p < 0.01$). We observe the same results with 100 and 1,000 clusters induced by k-means algorithm and with 500 clusters induced by hierarchical clustering from the previous analysis. Confirming the statistical significance, the histogram of KL divergence values shows that the label distributions in most of the clusters differ from the overall distribution of labels in the training set (Figure 5.4). This suggests that Skip-gram discourse vector representation might ease classification by projecting the discourse relations into the space such that most of the clusters are already partially classified.

5.6 Related work

Several efforts have been made to classify implicit discourse relations. Various external lexicons and resources have been used to capture the semantic representation of sentences (Pitler et al., 2009). Brown cluster pairs have been used to remedy the sparsity problem found in this kind of task when using a battery of conventional binary features (Rutherford and Xue, 2014). However, without additional data, the additional benefits of feature-engineering start to plateau. Our new neural-network-based method breaks the trend of creating more and more features, while still encapsulating the traditional paradigm.

Unsupervised or semi-supervised approaches seem suitable for this task but turn out to be limited in their efficacy (Marcu and Echihabi, 2002b). Sporleder and Lascarides (2008) use discourse connectives to label discourse relations automatically and train supervised models on these newly

labeled data. Unfortunately, these approaches fall short because explicit and implicit discourse relation differ linguistically.

Neural networks have recently seen successes in natural language and speech processing. Notably, a deep neural network architecture achieves the state-of-the-art in acoustic modeling (Hinton et al., 2012). A recursive neural tensor network has been used to model semantic compositionality for fine-grained sentiment classification tasks (Socher et al., 2013b; Socher et al., 2012). To the best of our knowledge, the recurrent convolutional neural network proposed by Kalchbrenner and Blunsom (2013) is the only neural network model applied for discourse analysis. This class of model convolves distributed representation of words into an utterance vector and a discourse context vector, which can be used for dialogue act classification. We do not use a recurrent architecture because written text does not have a turn-taking nature usually observed in a dialogue. But we are not aware of any neural discourse models for text or monologue.

5.7 Conclusion

We present the Neural Discourse Mixture Model, which mixes the continuous-valued Skip-gram representation of a discourse relation with a large array of traditional binary values used by Naive Bayes classifiers. The NDMM achieves the performance gain over the current state-of-the-art system partly because the discourse vectors provide better representation of a discourse relation than Brown cluster pair representation.

Upon linguistic analysis, the Skip-gram discourse vectors capture some important aspects of the discourse that help distinguish between certain types of implicit discourse relation. As a classification model, the discourse vector network alone performs competitively for some labels. These results suggest that distributed discourse relation representation is a promising direction for improving implicit discourse relation classification.

Topic	Size	Examples
Medical	6	**Using small electrical shocks applied to her feet , they were able to monitor sensory nerves**. *The shocks generated nerve impulses that traveled via spine to brain and showed up clearly on a brain-wave monitor , indicating no damage to the delicate spinal tissue.* (WSJ0297) **The devices' most remarkable possibilities , though, involve the brain**. *Probing with the stimulators, National Institutes of Health scientists recently showed how the brain reorganizes motor-control resources after an amputation.* (WSJ0297) **It relies on the fact that certain atoms give off detectable signals when subjected to an intense magnetic field**. *It's the same phenomenon used in the new MRI (magnetic resonance imaging) scanners being used in hospitals in place of X-ray scans.* (WSJ1218)
Tax	7	**But they also are to see that taxpayers get all allowable tax benefits and to ask if filers who sought IRS aid were satisfied with it**. *Courts have ruled that taxpayers must submit to TCMP audits , but the IRS will excuse from the fullscale rigors anyone who was audited without change for either 1986 or 1987.* (WSJ0293) **Many openings for mass cheating, such as questionable tax shelters and home offices, have gaped so broadly that Congress has passed stringent laws to close them**. *Deductions of charitable gifts of highly valued art now must be accompanied by appraisals.* (WSJ1570) **For corporations , the top tax rate on the sale of assets held for more than three years would be cut to 33 % from the current top rate of 34 %** *That rate would gradually decline to as little as 29 % for corporate assets held for 15 years.* (WSJ1869)
Broadcasting	28	**It isn't clear how much those losses may widen because of the short Series**. *Had the contest gone a full seven games , ABC could have reaped an extra $ 10 million in ad sales on the seventh game alone , compared with the ad take it would have received for regular prime-time shows.* (WSJ0443) **The ads are just the latest evidence of how television advertising is getting faster on the draw**. *While TV commercials typically take weeks to produce, advertisers in the past couple of years have learned to turn on a dime, to crash out ads in days or even hours.* (WSJ0453) **And only a handful of small U.S. companies are engaged in high-definition software development**. It's estimated *that just about 250 hours of HD programming is currently available for airing.* (WSJ1386)

Table 5.3: Clustering analysis of Skip-gram discourse vector model reveals that the absence of topic shift within a discourse relation might provide some clues to the types of discourse relations. Arg1 and Arg2 are indicated by bold-faced text and italicized text respectively.

Chapter 6

Recurrent Neural Network for Discourse Analysis

The discourse structure of natural language has been analyzed and conceptualized under various frameworks (Mann and Thompson, 1988; Lascarides and Asher, 2007; Prasad et al., 2008). The Penn Discourse TreeBank (PDTB) and the Chinese Discourse Treebank (CDTB), currently the largest corpora annotated with discourse structures in English and Chinese respectively, view the discourse structure of a text as a set of discourse relations (Prasad et al., 2008; Zhou and Xue, 2012). Each discourse relation is grounded by a discourse connective taking two text segments as arguments (Prasad et al., 2008). Implicit discourse relations are those where discourse connectives are omitted from the text and yet the discourse relations still hold.

Neural network models are an attractive alternative for this task for at least two reasons. First, they can model the argument of an implicit discourse relation as dense vectors and suffer less from the data sparsity problem that is typical of the traditional feature engineering paradigm. Second, they should be easily extended to other languages as they do not require human-annotated lexicons. However, despite the many nice properties of neural network models, it is not clear how well they will fare with a small dataset, typically found in discourse annotation projects. Moreover, it is

not straightforward to construct a single vector that properly represents the "semantics" of the arguments. As a result, neural network models that use dense vectors have been shown to have inferior performance against traditional systems that use manually crafted features, unless the dense vectors are combined with the hand-crafted surface features (Ji and Eisenstein, 2015).

In this work, we explore multiple neural architectures in an attempt to find the best distributed representation and neural network architecture suitable for this task in both English and Chinese. We do this by probing the different points on the spectrum of structurality from structureless bag-of-words models to sequential and tree-structured models. We use feedforward, sequential long short-term memory (LSTM), and tree-structured LSTM models to represent these three points on the spectrum. To the best of our knowledge, there is no prior study that investigates the contribution of the different architectures in neural discourse analysis.

Our main contributions and findings from this work can be summarized as follows:

- Our neural discourse model performs comparably with or even outperforms systems with surface features across different fine-grained discourse label sets.
- We investigate the contribution of the linguistic structures in neural discourse modeling and found that high-dimensional word vectors trained on a large corpus can compensate for the lack of structures in the model, given the small amount of annotated data.
- We found that modeling the interaction across arguments via hidden layers is essential to improving the performance of an implicit discourse relation classifier.
- We present the first neural CDTB-style Chinese discourse parser, confirming that our current results and other previous findings conducted on English data also hold cross-linguistically.

6.1 Related Work

The prevailing approach for this task is to use surface features derived from various semantic lexicons (Pitler et al., 2009), reducing the number of parameters by mapping raw word tokens in the arguments of discourse relations to a limited number of entries in a semantic lexicon such as polarity and verb classes.

Along the same vein, Brown cluster assignments have also been used as a general purpose lexicon that requires no human manual annotation (Rutherford and Xue, 2014). However, these solutions still suffer from the data sparsity problem and almost always require extensive feature selection to work well (Park and Cardie, 2012; Lin et al., 2009; Ji and Eisenstein, 2015). The work we report here explores the use of the expressive power of distributed representations to overcome the data sparsity problem found in the traditional feature engineering paradigm.

Neural network modeling has attracted much attention in the NLP community recently and has been explored to some extent in the context of this task. Recently, Braud and Denis (2015) tested various word vectors as features for implicit discourse relation classification and show that distributed features achieve the same level of accuracy as one-hot representations in some experimental settings. Ji et al. (2015; 2016) advance the state of the art for this task using recursive and recurrent neural networks. In the work we report here, we systematically explore the use of different neural network architectures and show that when high-dimensional word vectors are used as input, a simple feed-forward architecture can outperform more sophisticated architectures such as sequential and tree-based LSTM networks, given the small amount of data.

Recurrent neural networks, especially LSTM networks, have changed the paradigm of deriving distributed features from a sentence (Hochreiter and Schmidhuber, 1997), but they have not been much explored in the realm of discourse parsing. LSTM models have been notably used to encode the meaning of source language sentence in neural machine translation (Cho et al., 2014; Devlin et al., 2014) and recently used to encode the meaning of an entire sentence to be used as features (Kiros et al., 2015). Many neural architectures have been explored and evaluated, but there is no

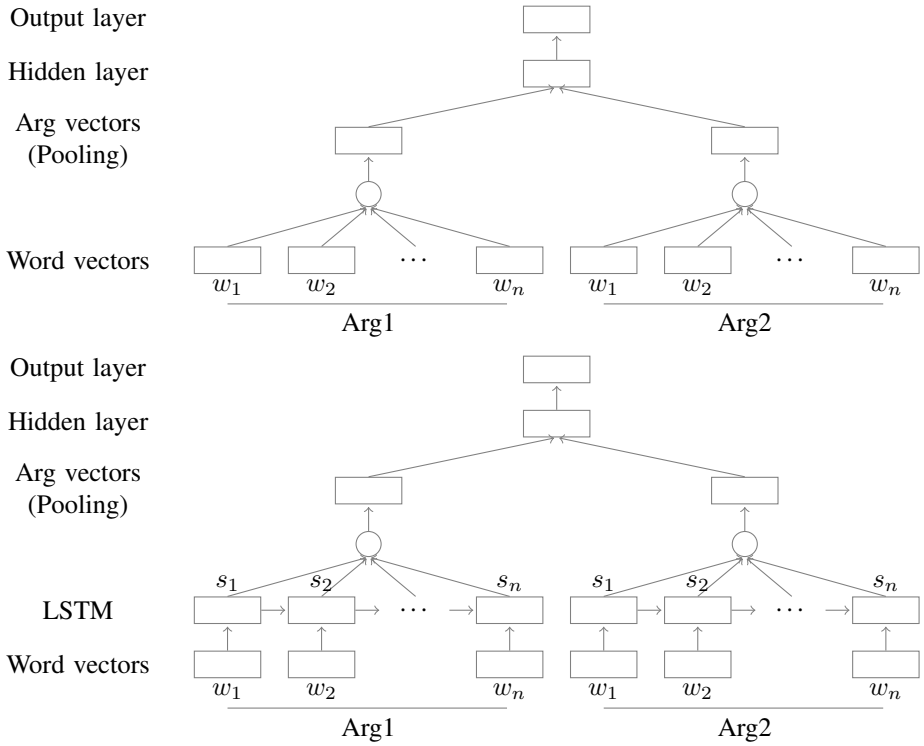

Figure 6.1: (Top) Feedforward architecture. (Bottom) Sequential Long Short-Term Memory architecture.

single technique that is decidedly better across all tasks. The LSTM-based models such as Kiros et al. (2015) perform well across tasks but do not outperform some other strong neural baselines. Ji et al. (2016) uses a joint discourse language model to improve the performance on the coarse-grained label in the PDTB, but in our case, we would like to deduce how well LSTM fares in fine-grained implicit discourse relation classification. A joint discourse language model might not scale well to finer-grained label set, which is more practical for application.

6.2 Model Architectures

Following previous work, we assume that the two arguments of an implicit discourse relation are given so that we can focus on predicting the senses of the implicit discourse relations. The input

to our model is a pair of text segments called Arg1 and Arg2, and the label is one of the senses defined in the Penn Discourse Treebank as in the example below:

Input:

Arg1 Senator Pete Domenici calls this effort "the first gift of democracy"

Arg2 The Poles might do better to view it as a Trojan Horse.

Output:

Sense Comparison.Contrast

In all architectures, each word in the argument is represented as a k-dimensional word vector trained on an unannotated data set. We use various model architectures to transform the semantics represented by the word vectors into distributed continuous-valued features. In the rest of the section, we explain the details of the neural network architectures that we design for the implicit discourse relations classification task. The models are summarized schematically in Figure 6.1.

6.2.1 Bag-of-words Feedforward Model

This model does not model the structure or word order of a sentence. The features are simply obtained through element-wise pooling functions. Pooling is one of the key techniques in neural network modeling of computer vision (Krizhevsky et al., 2012; LeCun et al., 2010). Max pooling is known to be very effective in vision, but it is unclear what pooling function works well when it comes to pooling word vectors. Summation pooling and mean pooling have been claimed to perform well at composing meaning of a short phrase from individual word vectors (Le and Mikolov, 2014; Blacoe and Lapata, 2012; Mikolov et al., 2013b; Braud and Denis, 2015). The Arg1 vector a^1 and Arg2 vector a^2 are computed by applying element-wise pooling function f on all of the N_1 word vectors in Arg1 $w^1_{1:N_1}$ and all of the N_2 word vectors in Arg2 $w^2_{1:N_2}$ respectively:

$$a^1_i = f(w^1_{1:N_1,i})$$
$$a^2_i = f(w^2_{1:N_2,i})$$

We consider three different pooling functions namely max, summation, and mean pooling functions:

$$\begin{aligned} f_{max}(w_{1:N}, i) &= \max_{j=1}^{N} w_{j,i} \\ f_{sum}(w_{1:N}, i) &= \sum_{j=1}^{N} w_{j,i} \\ f_{mean}(w_{1:N}, i) &= \sum_{j=1}^{N} w_{j,i}/N \end{aligned}$$

Inter-argument interaction is modeled directly by the hidden layers that take argument vectors as features. Discourse relations cannot be determined based on the two arguments individually. Instead, the sense of the relation can only be determined when the arguments in a discourse relation are analyzed jointly. The first hidden layer h_1 is the non-linear transformation of the weighted linear combination of the argument vectors:

$$h_1 = \tanh(W_1 \cdot a^1 + W_2 \cdot a^2 + b_{h_1})$$

where W_1 and W_2 are $d \times k$ weight matrices and b_{h_1} is a d-dimensional bias vector. Further hidden layers h_t and the output layer o follow the standard feedforward neural network model.

$$\begin{aligned} h_t &= \tanh(W_{h_t} \cdot h_{t-1} + b_{h_t}) \\ o &= \text{softmax}(W_o \cdot h_T + b_o) \end{aligned}$$

where W_{h_t} is a $d \times d$ weight matrix, b_{h_t} is a d-dimensional bias vector, and T is the number of hidden layers in the network.

6.2.2 Sequential Long Short-Term Memory (LSTM)

A sequential Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) models the semantics of a sequence of words through the use of hidden state vectors. Therefore, the word ordering

does affect the resulting hidden state vectors, unlike the bag-of-word model. For each word vector at word position t, we compute the corresponding hidden state vector s_t and the memory cell vector c_t from the previous step.

$$\begin{aligned} i_t &= \text{sigmoid}(W_i \cdot w_t + U_i \cdot s_{t-1} + b_i) \\ f_t &= \text{sigmoid}(W_f \cdot w_t + U_f \cdot s_{t-1} + b_f) \\ o_t &= \text{sigmoid}(W_o \cdot w_t + U_o \cdot s_{t-1} + b_o) \\ c'_t &= \tanh(W_c \cdot w_t + U_c \cdot s_{t-1} + b_c) \\ c_t &= c'_t * i_t + c_{t-1} * f_t \\ s_t &= c_t * o_t \end{aligned}$$

where $*$ is elementwise multiplication. The argument vectors are the results of applying a pooling function over the hidden state vectors.

$$a_i^1 = f(s_{1:N_1,i}^1)$$
$$a_i^2 = f(s_{1:N_2,i}^2)$$

In addition to the three pooling functions that we describe in the previous subsection, we also consider using only the last hidden state vector, which should theoretically be able to encode the semantics of the entire word sequence.

$$f_{last}(s_{1:N,i}) = s_{N,i}$$

Inter-argument interaction and the output layer are modeled in the same fashion as the bag-of-words model once the argument vector is computed.

6.2.3 Tree LSTM

The principle of compositionality leads us to believe that the semantics of the argument vector should be determined by the syntactic structures and the meanings of the constituents. For a fair

Sense	Train	Dev	Test
Comparison.Concession	192	5	5
Comparison.Contrast	1612	82	127
Contingency.Cause	3376	120	197
Contingency.Pragmatic cause	56	2	5
Expansion.Alternative	153	2	15
Expansion.Conjunction	2890	115	116
Expansion.Instantiation	1132	47	69
Expansion.List	337	5	25
Expansion.Restatement	2486	101	190
Temporal.Asynchronous	543	28	12
Temporal.Synchrony	153	8	5
Total	12930	515	766

Table 6.1: The distribution of the level 2 sense labels in the Penn Discourse Treebank. The instances annotated with two labels are not double-counted (only first label is counted here), and partial labels are excluded.

comparison with the sequential model, we apply the same formulation of LSTM on the binarized constituent parse tree. The hidden state vector now corresponds to a constituent in the tree. These hidden state vectors are then used in the same fashion as the sequential LSTM. The mathematical formulation is the same as Tai et al. (2015).

This model is similar to the recursive neural networks proposed by Ji and Eisenstein (2015). Our model differs from their model in several ways. We use the LSTM networks instead of the "vanilla" RNN formula and expect better results due to less complication with vanishing and exploding gradients during training. Furthermore, our purpose is to compare the influence of the model structures. Therefore, we must use LSTM cells in both sequential and tree LSTM models for a fair and meaningful comparison. The more in-depth comparison of our work and recursive neural network model by Ji and Eisenstein (2015) is provided in the discussion section.

6.3 Corpora and Implementation

The Penn Discourse Treebank (PDTB) We use the PDTB due to its theoretical simplicity in discourse analysis and its reasonably large size. The annotation is done as another layer on

the Penn Treebank on Wall Street Journal sections. Each relation consists of two spans of text that are minimally required to infer the relation, and the sense is organized hierarchically. The classification problem can be formulated in various ways based on the hierarchy. Previous work in this task has been done over three schemes of evaluation: top-level 4-way classification (Pitler et al., 2009), second-level 11-way classification (Lin et al., 2009; Ji and Eisenstein, 2015), and modified second-level classification introduced in the CoNLL 2015 Shared Task (Xue et al., 2015). We focus on the second-level 11-way classification because the labels are fine-grained enough to be useful for downstream tasks and also because the strongest neural network systems are tuned to this formulation. If an instance is annotated with two labels (∼3% of the data), we only use the first label. Partial labels, which constitute ∼2% of the data, are excluded. Table **??** shows the distribution of labels in the training set (sections 2-21), development set (section 22), and test set (section 23).

Training Weight initialization is uniform random, following the formula recommended by Bengio (2012). The cost function is the standard cross-entropy loss function, as the hinge loss function (large-margin framework) yields consistently inferior results. We use Adagrad as the optimization algorithm of choice. The learning rates are tuned over a grid search. We monitor the accuracy on the development set to determine convergence and prevent overfitting. L2 regularization and/or dropout do not make a big impact on performance in our case, so we do not use them in the final results.

Implementation All of the models are implemented in Theano (Bergstra et al., 2010; Bastien et al., 2012). The gradient computation is done with symbolic differentiation, a functionality provided by Theano. Feedforward models and sequential LSTM models are trained on CPUs on Intel Xeon X5690 3.47GHz, using only a single core per model. A tree LSTM model is trained on a GPU on Intel Xeon CPU E5-2660. All models converge within hours.

Model	Accuracy
PDTB Second-level senses	
Most frequent tag baseline	25.71
Our best tree LSTM	34.07
Ji & Eisenstein, (2015)	36.98
Our best sequential LSTM variant	38.38
Our best feedforward variant	39.56
Lin et al., (2009)	40.20

Table 6.2: Performance comparison across different models for second-level senses.

6.4 Experiment on the Second-level Sense in the PDTB

We want to test the effectiveness of the inter-argument interaction and the three models described above on the fine-grained discourse relations in English. The data split and the label set are exactly the same as previous works that use this label set (Lin et al., 2009; Ji and Eisenstein, 2015).

Preprocessing All tokenization is taken from the gold standard tokenization in the PTB (Marcus et al., 1993). We use the Berkeley parser to parse all of the data (Petrov et al., 2006). We test the effects of word vector sizes. 50-dimensional and 100-dimensional word vectors are trained on the training sections of WSJ data, which is the same text as the PDTB annotation. Although this seems like too little data, 50-dimensional WSJ-trained word vectors have previously been shown to be the most effective in this task (Ji and Eisenstein, 2015). Additionally, we also test the off-the-shelf word vectors trained on billions of tokens from Google News data freely available with the `word2vec` tool. All word vectors are trained on the Skip-gram architecture (Mikolov et al., 2013b; Mikolov et al., 2013a). Other models such as GloVe and continuous bag-of-words seem to yield broadly similar results (Pennington et al., 2014). We keep the word vectors fixed, instead of fine-tuning during training.

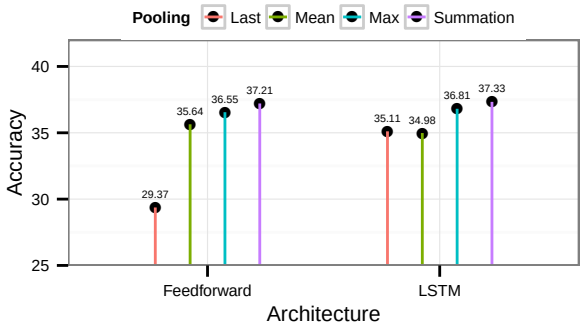

Figure 6.2: Summation pooling gives the best performance in general. The results are shown for the systems using 100-dimensional word vectors and one hidden layer.

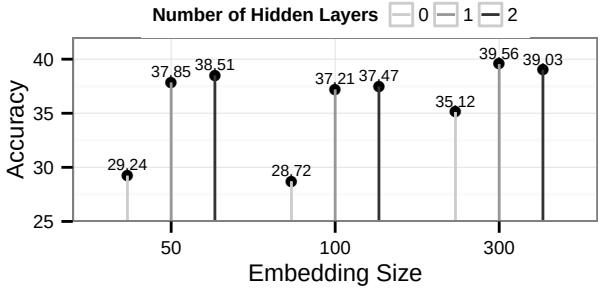

Figure 6.3: Inter-argument interaction can be modeled effectively with hidden layers. The results are shown for the feedforward models with summation pooling, but this effect can be observed robustly in all architectures we consider.

6.4.1 Results and discussion

The feedforward model performs best overall among all of the neural architectures we explore (Table 6.2). It outperforms the recursive neural network with bilinear output layer introduced by Ji and Eisenstein (2015) ($p < 0.05$; bootstrap test) and performs comparably with the surface feature baseline (Lin et al., 2009), which uses various lexical and syntactic features and extensive feature selection. Tree LSTM achieves inferior accuracy than our best feedforward model. The best configuration of the feedforward model uses 300-dimensional word vectors, one hidden layer, and the summation pooling function to derive argument feature vectors. The model behaves well during training and converges in less than an hour on a CPU.

The sequential LSTM model outperforms the feedforward model when word vectors are not

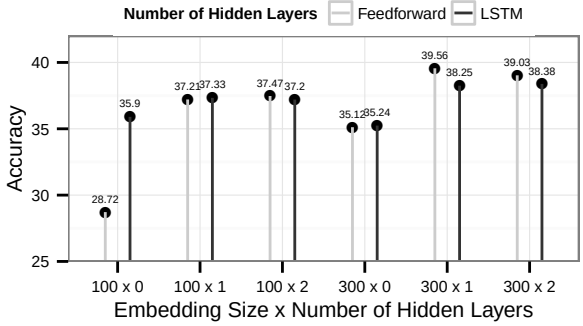

Figure 6.4: Comparison between feedforward and sequential LSTM when using summation pooling function.

		No hidden layer				1 hidden layer				2 hidden layers			
Architecture	k	max	mean	sum	last	max	mean	sum	last	max	mean	sum	last
Feedforward	50	31.85	31.98	29.24	-	33.28	34.98	37.85	-	34.85	35.5	38.51	-
LSTM	50	31.85	32.11	34.46	31.85	34.07	33.15	36.16	34.34	36.16	35.11	37.2	35.24
Tree LSTM	50	28.59	28.32	30.93	28.72	29.89	30.15	32.5	31.59	32.11	31.2	32.5	29.63
Feedforward	100	33.29	32.77	28.72	-	36.55	35.64	37.21	-	36.55	36.29	37.47	-
LSTM	100	30.54	33.81	35.9	33.02	36.81	34.98	37.33	35.11	37.46	36.68	37.2	35.77
Tree LSTM	100	29.76	28.72	31.72	31.98	31.33	26.89	33.02	33.68	32.63	31.07	32.24	33.02
Feedforward	300	32.51	34.46	35.12	-	35.77	38.25	**39.56**	-	35.25	38.51	39.03	-
LSTM	300	28.72	34.59	35.24	34.64	38.25	36.42	37.07	35.5	**38.38**	37.72	37.2	36.29
Tree LSTM	300	28.45	31.59	32.76	26.76	33.81	32.89	33.94	32.63	32.11	32.76	**34.07**	32.50

Table 6.3: Compilation of all experimental configurations for 11-way classification on the PDTB test set. k is the word vector size. Bold-faced numbers indicate the best performance for each architecture, which is also shown in Table 6.2.

high-dimensional and not trained on a large corpus (Figure 6.4). Moving from 50 units to 100 units trained on the same dataset, we do not observe much of a difference in performance in both architectures, but the sequential LSTM model beats the feedforward model in both settings. This suggests that only 50 dimensions are needed for the WSJ corpus. However, the trend reverses when we move to 300-dimensional word vectors trained on a much larger corpus. These results suggest an interaction between the lexical information encoded by word vectors and the structural information encoded by the model itself.

Hidden layers, especially the first one, make a substantial impact on performance. This effect is observed across all architectures (Figure 6.3). Strikingly, the improvement can be as high as 8% absolute when used with the feedforward model with small word vectors. We tried up to four

hidden layers and found that the additional hidden layers yield diminishing—if not negative—returns. These effects are not an artifact of the training process as we have tuned the models quite extensively, although it might be the case that we do not have sufficient data to fit those extra parameters.

Summation pooling is effective for both feedforward and LSTM models (Figure 6.2). The word vectors we use have been claimed to have some additive properties (Mikolov et al., 2013b), so summation pooling in this experiment supports this claim. Max pooling is only effective for LSTM, probably because the values in the word vector encode the abstract features of each word relative to each other. It can be trivially shown that if all of the vectors are multiplied by -1, then the results from max pooling will be totally different, but the word similarities remain the same. The memory cells and the state vectors in the LSTM models transform the original word vectors to work well the max pooling operation, but the feedforward net cannot transform the word vectors to work well with max pooling as it is not allowed to change the word vectors themselves.

6.4.2 Discussion

Why does the feedforward model outperform the LSTM models? Sequential and tree LSTM models might work better if we are given larger amount of data. We observe that LSTM models outperform the feedforward model when word vectors are smaller, so it is unlikely that we train the LSTMs incorrectly. It is more likely that we do not have enough annotated data to train a more powerful model such as LSTM. In previous work, LSTMs are applied to tasks with a lot of labeled data compared to mere 12,930 instances that we have (Vinyals et al., 2015; Chiu and Nichols, 2015; İrsoy and Cardie, 2014). Another explanation comes from the fact that the contextual information encoded in the word vectors can compensate for the lack of structure in the model in this task. Word vectors are already trained to encode the words in their linguistic context especially information from word order.

Our discussion would not be complete without explaining our results in relation to the recursive

neural network model proposed by Ji and Eisenstein (2015). Why do sequential LSTM models outperform recursive neural networks or tree LSTM models? Although this first comes as a surprise to us, the results are consistent with recent works that use sequential LSTM to encode syntactic information. For example, Vinyals et al. (2015) use sequential LSTM to encode the features for syntactic parse output. Tree LSTM seems to show improvement when there is a need to model long-distance dependency in the data (Tai et al., 2015; Li et al., 2015). Furthermore, the benefits of tree LSTM are not readily apparent for a model that discards the syntactic categories in the intermediate nodes and makes no distinction between heads and their dependents, which are at the core of syntactic representations.

Another point of contrast between our work and Ji and Eisenstein's (2015) is the modeling choice for inter-argument interaction. Our experimental results show that the hidden layers are an important contributor to the performance for all of our models. We choose linear inter-argument interaction instead of bilinear interaction, and this decision gives us at least two advantages. Linear interaction allows us to stack up hidden layers without the exponential growth in the number of parameters. Secondly, using linear interaction allows us to use high dimensional word vectors, which we found to be another important component for the performance. The recursive model by Ji and Eisenstein (2015) is limited to 50 units due to the bilinear layer. Our choice of linear inter-argument interaction and high-dimensional word vectors turns out to be crucial to building a competitive neural network model for classifying implicit discourse relations.

6.5 Extending the results across label sets and languages

Do our feedforward models perform well without surface features across different label sets and languages as well? We want to extend our results to another label set and language by evaluating our models on non-explicit discourse relation data used in English and Chinese CoNLL 2016

Shared Task. We will have more confidence in our model if it works well across label sets. It is also important that our model works cross-linguistically because other languages might not have resources such as semantic lexicons or parsers, required by some previously used features.

6.5.1 English discourse relations

We follow the experimental setting used in CoNLL 2015-2016 Shared Task as we want to compare our results against previous systems. This setting differs from the previous experiment in a few ways. Entity relations (EntRel) and alternative lexicalization relations (AltLex) are included in this setting. The label set is modified by the shared task organizers into 15 different senses including EntRel as another sense (Xue et al., 2015). We use the 300-dimensional word vector used in the previous experiment and tune the number of hidden layers and hidden units on the development set. The best results from last year's shared task are used as a strong baseline. It only uses surface features and also achieves the state-of-the-art performance under this label set (Wang and Lan, 2015). These features are similar to the ones used by Lin et al. (2009).

6.5.2 Chinese discourse relations

We evaluate our model on the Chinese Discourse Treebank (CDTB) because its annotation is the most comparable to the PDTB (Zhou and Xue, 2015). The sense set consists of 10 different senses, which are not organized in a hierarchy, unlike the PDTB. We use the version of the data provided to the CoNLL 2016 Shared Task participants. This version has 16,946 instances of discourse relations total in the combined training and development sets. The test set is not yet available at the time of submission, so the system is evaluated based on the average accuracy over 7-fold cross-validation on the combined set of training and development sets.

There is no previously published baseline for Chinese. To establish baseline comparison, we use MaxEnt models loaded with the feature sets previously shown to be effective for English, namely dependency rule pairs, production rule pairs (Lin et al., 2009), Brown cluster pairs (Rutherford

Model	Acc.
CoNLL-ST 2015-2016 English	
Most frequent tag baseline	21.36
Our best LSTM variant	31.76
Wang and Lan (2015) - winning team	34.45
Our best feedforward variant	**36.26**
CoNLL-ST 2016 Chinese	
Most frequent tag baseline	77.14
ME + Production rules	80.81
ME + Dependency rules	82.34
ME + Brown pairs (1000 clusters)	82.36
Out best LSTM variant	82.48
ME + Brown pairs (3200 clusters)	82.98
ME + Word pairs	83.13
ME + All feature sets	84.16
Our best feedforward variant	**85.45**

Table 6.4: Our best feedforward variant significantly outperforms the systems with surface features ($p < 0.05$). ME=Maximum Entropy model

and Xue, 2014), and word pairs (Marcu and Echihabi, 2002b). We use information gain criteria to select the best subset of each feature set, which is crucial in feature-based discourse parsing.

Chinese word vectors are induced through CBOW and Skipgram architecture in `word2vec` (Mikolov et al., 2013a) on Chinese Gigaword corpus (Graff and Chen, 2005) using default settings. The number of dimensions that we try are 50, 100, 150, 200, 250, and 300. We induce 1,000 and 3,000 Brown clusters on the Gigaword corpus.

6.5.3 Results

Table 6.4 shows the results for the models which are best tuned on the number of hidden units, hidden layers, and the types of word vectors. The feedforward variant of our model significantly outperforms the strong baselines in both English and Chinese ($p < 0.05$ bootstrap test). This suggests that our approach is robust against different label sets, and our findings are valid across

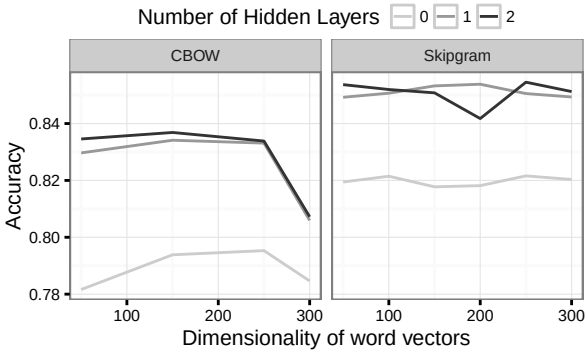

Figure 6.5: Comparing the accuracies across Chinese word vectors for feedforward model.

languages. Our Chinese model outperforms all of the feature sets known to work well in English despite using only word vectors.

The choice of neural architecture used for inducing Chinese word vectors turns out to be crucial. Chinese word vectors from Skipgram model perform consistently better than the ones from CBOW model (Figure 6.5). These two types of word vectors do not show much difference in the English tasks.

6.6 Conclusions and future work

We report a series of experiments that systematically probe the effectiveness of various neural network architectures for the task of implicit discourse relation classification. Given the small amount of annotated data, we found that a feedforward variant of our model combined with hidden layers and high dimensional word vectors outperforms more complicated LSTM models. Our model performs better or competitively against models that use manually crafted surface features, and it is the first neural CDTB-style Chinese discourse parser. We will make our code and models publicly available.

Chapter 7

Robust Light-weight Multilingual Neural Discourse Parser

7.1 Introduction

Most of the previous work in discourse relation classification are much limited in their applicability because they show their efficacy on specific label sets and only in English. The automatic discourse analysis needs to be adapted to the application in order to be useful. Surface features approaches are quite effective, but we cannot extend them to other languages where semantic lexicon and syntactic parsers are not available or not accurate (Lin et al., 2009; **?**). These approaches also require extensive feature selection to make them effective to a specific label set. Explicit discourse relation classification does not suffer from this problem as the discourse connectives determine the senses in the relation quite reliably. Non-explicit discourse relations, on the other hand, benefit more robust methods that work across many settings.

Neural network models address this problem indirectly by avoiding the semantic lexicons and relying word vectors as the data-driven semantic lexicon, where their features are abstract continuous values. However, the previous work in neural network model in task does require parses,

which bring us back to the extensibility problem we face before (Ji and Eisenstein, 2015). Recent neural approaches also integrate language modeling and considerably improve the results of both language and discourse modeling. It is unclear whether this approach can be extended to be other label sets other than the four top-level senses.

Why do we care about the robustness against varying label sets? Discourse analysis differs from other linguistic annotation in that there is no single correct sense inventory or annotation for discourse. In other words, discourse analysis depends much on the domain of application or studies. One sense inventory (label set) might be appropriate for one application, but not the others. Rhetorical Structure Theory (RST), for example, argues that their sense inventory is one of the many valid ones given this same theory (Mann and Thompson, 1988). The Penn Discourse Treebank (PDTB) organizes the labels hierarchically to allow some flexibility in analysis at various levels depending on the application. These suggest that a discourse parsing technique that is designed or tuned to one specific label set might not be as useful in real application.

Another concern with regard to robustness and applicability comes from the fact that discourse phenomena should be less language dependent than other aspects of linguistic analysis, such as syntactic or morphological analysis. For example, not every language has a case-marking system , but every language has a causal discourse relation with or without discourse connectives. That suggests that a good algorithm for discourse parsing must be applicable to many languages. From engineering point of view, we would like to extend what we know from English discourse analysis to other languages in a straightforward fashion if possible. Discourse relations are being annotated in the style of the PDTB, and multilingual discourse parsing has gained some attention within the recent years (Zhou and Xue, 2012; Oza et al., 2009; Al-Saif and Markert, 2010; Zeyrek and Webber, 2008). More notably, the Chinese Discourse Treebank (CDTB) has also been annotated with non-explicit discourse relations, unlike the other languages.

Here, we propose a light-weight yet robust non-explicit discourse relation classifier. The core of the model comes from word vectors and simple feedforward architecture presented in the previous

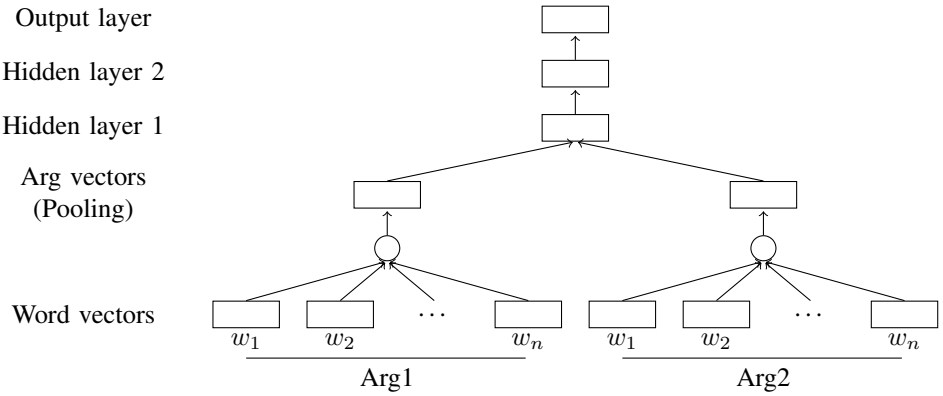

Figure 7.1: Model architecture

chapter. We show that our model performs better under many label sets than the models that use surface features derived from syntactic parses or semantic lexicon. Our model also outperforms such system in Chinese, presenting the first Chinese neural discourse parser. All of these are achieved by around 200k parameters and minutes of training on a CPU without automatic parses and other manually annotated resources.

7.2 Model description

The Arg1 vector a^1 and Arg2 vector a^2 are computed by applying element-wise pooling function f on all of the N_1 word vectors in Arg1 $w^1_{1:N_1}$ and all of the N_2 word vectors in Arg2 $w^2_{1:N_2}$ respectively:

$$a^1_i = \sum_{j=1}^{N} w^1_{j,i}$$

$$a^2_i = \sum_{j=1}^{N} w^2_{j,i}$$

Inter-argument interaction is modeled directly by the hidden layers that take argument vectors as features. Discourse relations cannot be determined based on the two arguments individually. Instead, the sense of the relation can only be determined when the arguments in a discourse relation

are analyzed jointly. The first hidden layer h_1 is a non-linear transformation of the weighted linear combination of the argument vectors:

$$h_1 = \tanh(W_1 \cdot a^1 + W_2 \cdot a^2 + b_{h_1})$$

where W_1 and W_2 are $d \times k$ weight matrices and b_{h_1} is a d-dimensional bias vector. Further hidden layers h_t and the output layer o follow the standard feedforward neural network model.

$$h_t = \tanh(W_{h_t} \cdot h_{t-1} + b_{h_t})$$
$$o = \text{softmax}(W_o \cdot h_T + b_o)$$

where W_{h_t} is a $d \times d$ weight matrix, b_{h_t} is a d-dimensional bias vector, and T is the number of hidden layers in the network.

This model has been studied more extensively in Chapter 6. We have experimented and tuned most components: pooling functions create the argument vectors, the type of word vectors, and the model architectures themselves. We found the model variant with two hidden layers and 300 hidden units to work well across many settings. The model has the total of around 270k parameters.

7.3 Experiments

7.3.1 Data

The English and Chinese are taken from the Penn Discourse Treebank and the Chinese Discourse Treebank respectively (Prasad et al., 2008; **?**). We use all non-explicit discourse relations, which include Entity Relation (EntRel) and Alternative Lexicalization (AltLex) in addition to implicit discourse relations. And we use the sense inventory used in the CoNLL shared task 2015 - 2016 (Xue et al., 2015), which differs slightly from the settings that we have used in all previous chapters.

7.3.2 Word vectors

English word vectors are taken from 300-dimensional Skip-gram word vectors trained on Google News data, provided by the shared task organizers (Mikolov et al., 2013a; Xue et al., 2015). We trained our own 250-dimensional Chinese word vectors on Gigaword corpus, which is the same corpus used by the 300-dimensional Chinese word vectors provided by the shared task organizers (Graff and Chen, 2005). We found the 250-dimensional version to work better despite fewer parameters.

7.3.3 Training

Weight initialization is uniform random, following the formula recommended by Bengio (2012). Word vectors are fixed during training. The cost function is the standard cross-entropy loss function, and we use Adagrad as the optimization algorithm of choice. We monitor the accuracy on the development set to determine convergence.

7.3.4 Implementation

All of the models are implemented in Theano (Bergstra et al., 2010; Bastien et al., 2012). The gradient computation is done with symbolic differentiation, a functionality provided by Theano. The models are trained on CPUs on Intel Xeon X5690 3.47GHz, using only a single core per model. The models converge in minutes. The implementation, the training script, and the trained model are already made available [1] .

7.3.5 Baseline

The winning system from last year's task serves as a strong baseline for English. We choose this system because it represents one of the strongest systems that utilizes exclusively surface features

[1] https://github.com/attapol/nn_discourse_parser

and extensive semantic lexicon (Wang and Lan, 2015). This approach uses a MaxEnt model loaded with millions of features.

We use Brown cluster pair features as the baseline for Chinese as there is no previous system for Chinese. We use 3200 clusters to create features and perform feature selection on the development set based on the information gain criteria. We end up with 10,000 features total.

	Development set		Test set		Blind test set	
Sense	Baseline	Ours	Baseline	Ours	Baseline	Ours
Comparison.Concession	0	0	0	0	0	0
Comparison.Contrast	0.098	0.1296	0.1733	0.1067	0	0
Contingency.Cause.Reason	0.4398	0.3514	0.3621	0.4	0.2878	0.3103
Contingency.Cause.Result	0.2597	0.1951	0.1549	0.1722	0.2254	0.1818
EntRel	0.6247	0.5613	0.5265	0.4892	0.5471	0.5516
Expansion.Alternative.Chosen alternative	0	0	0	0	0	0
Expansion.Conjunction	**0.4591**	0.3874	**0.3068**	0.2468	**0.3154**	0.2644
Expansion.Instantiation	0.2105	**0.4051**	0.3261	**0.4962**	0.1633	**0.25**
Expansion.Restatement	0.3482	0.3454	0.2923	0.3483	0.3232	0.2991
Temporal.Asynchronous.Precedence	0	0.0714	0	0	0	0.125
Temporal.Asynchronous.Succession	0	0	0	0	0	0
Temporal.Synchrony	0	0	0	0	0	0
Accuracy	0.4331	0.4032	0.3455	0.3613	0.3629	0.3767
Most-frequent-tag Acc.	0.2320		0.2844		0.2136	

Table 7.1: F_1 scores for English non-explicit discourse relation. The bold-faced numbers highlight the senses where the classification of our model and the baseline model might be complementary.

7.4 Error Analysis

Comparing confusion matrices from the two approaches help us understand further what neural networks have achieved. We approximate Bayes Factors with uniform prior for each sense pair (c_i, c_j) for gold standard g and system p:

$$\frac{P(p = c_i, g = c_j)}{P(p = c_i)P(g = c_j)}$$

	Development set		Test set		Blind test set	
Sense	Baseline	Ours	Baseline	Ours	Baseline	Ours
Alternative	0	0	0	0	0	0
Causation	0	0	0	0	0	0
Conditional	0	0	0	0	0	0
Conjunction	0.7830	0.7928	0.7911	0.8055	0.7875	0.7655
Contrast	0	0	0	0	0	0
EntRel	0.4176	0.4615	0.5175	0.5426	0.0233	0.0395
Expansion	0.4615	0.4167	0.2333	0.4333	0.2574	0.5104
Purpose	0	0	0	0	0	0
Temporal	0	0	0	0	0	0
Accuracy	0.6634	0.683	0.6657	0.7047	0.6437	0.6338
Most-frequent-tag Acc.	0.6176		0.6351		0.7914	

Table 7.2: Sense-wise F_1 scores for Chinese non-explicit discourse relation.

	The true sense is ...				
, confused as ... by ...	Instantiation	Contrast	Result	Precedence	Synchrony
Conjunction		+		#+	#+
Restatement	+				
Result		#			#
Reason			+		

Table 7.3: Confusion pairs made by our neural network (#) and the baseline surface features (+) in English.

We tabulate all significant confusion pairs (i.e. Bayes Factor greater than a cut-off) made by each of the systems (Table 7.3). This is done on the development set only.

The distribution of the confusion pairs suggest that neural network and surface feature systems complement each other in some way. We see that the two systems only share two confusion pairs in common.

Temporal.Asynchronous senses are confused with Conjunction by both systems. Temporal senses are difficult to classify in implicit discourse relations since the annotation itself can be quite ambiguous. Expansion.Instantiation relations are misclassified as Expansion.Restatement

by surface feature systems. Neural network system performs better on Expansion.Instantiation than surface feature systems probably because neural network system can tease apart Expansion.Instantiation and Expansion.Restatement.

7.5 Conclusions

We present a light-weight neural network model, which is easy to deploy, retrain, and adapt to other languages and label sets. The model only needs word vectors trained on large corpora. Our approach performs competitively if not better than traditional systems with surface features and MaxEnt model despite having one or two orders of magnitude fewer parameters.

Chapter 8

Conclusions and Future Work

Over the course of this dissertation, we have explored the task of implicit discourse relation classification in various directions. We have developed Brown cluster pair features, new features under the traditional feature-engineering paradigm. These features address the problem of feature sparsity and also capture the inter-argument interaction, which is characteristic of discourse phenomena. We have also exploited signals from discourse connectives and their potential power in providing us with more training data. And as recent successes of neural network modeling have surfaced in the natural language processing community, we developed a neural discourse parser and showed that simple neural architecture can perform comparably or better than more complicated architecture while maintaining its simplicity with which one can replicate the results, deploy the systems, and more importantly adapt the model to different label sets and languages.

The task of implicit discourse relation classification and discourse parsing in general has gained much attention as a body of research has provided the foundation for the basic paradigm for computational approaches. We have organized a shared task, in which researchers are invited to work on this task over the same period of time. Many research groups around the world have participated in our endeavor. We hope that such endeavor will keep the research momentum going and lead to further understanding and improvement in this task.

Implicit discourse relation classification deserves more significant improvement as the previous evaluation has suggested. Here, we propose future research directions based on what we have learned as a research community and based on the work presented previously.

8.1 Distant Supervision with Neural Network

The success stories that we see with neural network in other domains mostly involve large amount of data obtained through expert annotation, crowdsourcing, or automatically. We indeed need a lot more data than what is available now. It is difficult or even unfeasible to detect causal relations between spans of text because it requires some world knowledge of cause and effects. We cannot manually annotate enough data for discourse without incurring exorbitant cost. This is the motivation for distant supervision for discourse parsing.

Future efforts must be made to obtain more training data automatically based on the existing data and existing neural models. Our distant supervision work in Chapter 4 has suggested that this paradigm is plausible, but it needs much better criteria for data selection. Neural networks are shown to be very effective discriminative models, so we can harness their power to harvest more data. For example, one might build a classifier directly to harvest more data that similar enough to the implicit discourse relations annotated in the PDTB.

Another approach along the same vein involves perturbing the data to make the models more robust. This technique is quite standard in preprocessing the data for computer vision. The perturbation artificially expands the dataset and make the models learn how to distinguish noise better. Essentially, we inject small noise without changing the labels to make the learning process focus better on the signals for the labels. However, it is unclear how one perturbs discourse data for this purpose. This is a research direction that deserves investigation.

8.2 Neural Network Model as a Cognitive Model

We hardly scratch the surface of what neural discourse parser can do. One of the strengths of this class of model is its capacity to model complicated processes explicitly. The progress within this task, however, is limited by our knowledge of how humans process and comprehend discourse. Linguistic theories that deal with discourse-level phenomena are relatively few compared to other branches of linguistics. Neural network has mostly been used in NLP as a discriminative model without trying to model how human processes languages. Neural network models can be used as a cognitive model of discourse comprehension as they have been used for modeling cognitive facilities such as morphological learning (McClelland and Patterson, 2002), reinforcement learning (Riedmiller, 2005), motor program learning (Kohonen, 1990). This aspect is often overlooked as the convolutional network (Krizhevsky et al., 2012) in computer vision and deep neural network (Senior et al., 2015) in speech recognition show excellent classification accuracy without modeling human cognition. Neural caption generator hardly addresses how man perceives a picture, understands the context of it, and describes it with a language. This points to the need for more precise modeling, which hopefully will also improve the performance.

8.3 Discourse Analysis in Other Languages and Domains

Initial efforts have already been made to make a discourse parser work in more languages and different domains. We have previously made a somewhat strong claim that a theory in discourse analysis or an algorithm in discourse parsing should universally apply to most if not all languages as the phenomena do not depend on linguistic typology or phylogeny. This remains just a claim as we do not have strong enough empirical evidence to support it. It is unfortunate that implicit discourse relation annotation is only available in English and Chinese as far as We am concerned.

More annotation will obviously need to be conducted.

Discourse analysis covers monologue, dialogue, and multi-party conversation. We have limited the scope of our work to written text or monologue, but we believe that the approaches we developed can be applied to different discourse structures. Dialogue and conversation parsing grows more important over the past years as our mode of communication has shifted toward mobile texting, social media, and online forum. In these cases, one can ask similar questions. How does one parse a dialogue or multi-party conversation into multiple discourses or smaller discourse units. We hypothesize that the methods used in parsing text or monologue can be augmented with structured model to parse out the relationships between text messages.

8.4 Concluding Remarks

Discourse analysis is indeed a crucial next step in natural language processing, yet the theoretical grounds in this realm are underexplored. A functional discourse parser will open doors for many more applications of natural language understanding, dialogue management, machine translation, and more. It is our hope that we will formulate better theories or models that help us understand discourse phenomena and raise the performance of a discourse parser to the practical level.

Bibliography

[Al-Saif and Markert2010] Amal Al-Saif and Katja Markert. 2010. The leeds arabic discourse treebank: Annotating discourse connectives for arabic. In *LREC*.

[Bahdanau et al.2014] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

[Baroni et al.2014] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 238–247.

[Bastien et al.2012] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

[Bengio2012] Yoshua Bengio. 2012. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer.

[Bergstra et al.2010] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June. Oral Presentation.

[Bezanson et al.2012] Jeff Bezanson, Stefan Karpinski, Viral B Shah, and Alan Edelman. 2012. Julia: A fast dynamic language for technical computing. *arXiv preprint arXiv:1209.5145*.

[Biran and McKeown2013] Or Biran and Kathleen McKeown. 2013. Aggregated word pair features for implicit discourse relation disambiguation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 69–73. The Association for Computational Linguistics.

[Bird2006] Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics.

[Blacoe and Lapata2012] William Blacoe and Mirella Lapata. 2012. A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 546–556. Association for Computational Linguistics.

[Braud and Denis2015] Chloé Braud and Pascal Denis. 2015. Comparing word representations for implicit discourse relation classification. In *Empirical Methods in Natural Language Processing (EMNLP 2015)*.

[Brown et al.1992] Peter F Brown, Peter V deSouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n -gram models of natural language. *Computational Linguistics*, 18(4):467–479, December.

[Carlson et al.2003] Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2003. *Building a discourse-tagged corpus in the framework of rhetorical structure theory*. Springer.

[Chang and Lin2011] Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Chen and Manning2014] Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 740–750.

[Chiu and Nichols2015] Jason PC Chiu and Eric Nichols. 2015. Named entity recognition with bidirectional lstm-cnns. *arXiv preprint arXiv:1511.08308*.

[Cho et al.2014] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

[De Marneffe et al.2006] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

[Devlin et al.2014] Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1370–1380.

[Duchi et al.2011a] John Duchi, Elad Hazan, and Yoram Singer. 2011a. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

[Duchi et al.2011b] John Duchi, Elad Hazan, and Yoram Singer. 2011b. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 999999:2121–2159.

[Elman1990] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

[Erhan et al.2010] Dumitru Erhan, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. Why does unsupervised pre-training help deep learning? *The Journal of Machine Learning Research*, 11:625–660.

[Feng and Hirst2012] Vanessa Wei Feng and Graeme Hirst. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 60–68. Association for Computational Linguistics.

[Feng and Hirst2014] Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of The 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Baltimore, USA, June*.

[Finkel et al.2005] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

[Fraser1996] Bruce Fraser. 1996. Pragmatic markers. *Pragmatics*, 6:167–190.

[Fraser2006] Bruce Fraser. 2006. Towards a theory of discourse markers. *Approaches to discourse particles*, 1:189–204.

[Gers et al.2000] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471.

[Graff and Chen2005] David Graff and Ke Chen. 2005. Chinese gigaword. *LDC Catalog No.: LDC2003T09, ISBN*, 1:58563–58230.

[Graff et al.2007] D Graff, J Kong, K Chen, and K Maeda. 2007. English gigaword third edition, 2007, ldc 2007t07.

[Hinton et al.2006] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554.

[Hinton et al.2012] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[Hutchinson2005] Ben Hutchinson. 2005. Modelling the similarity of discourse connectives. In *Proceedings of the the 27th Annual Meeting of the Cognitive Science Society (CogSci2005)*.

[İrsoy and Cardie2014] Ozan İrsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 720–728.

[Jacobs et al.1991] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

[Japkowicz2000] Nathalie Japkowicz. 2000. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI workshop on learning from imbalanced data sets*, volume 68.

[Ji and Eisenstein2014] Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 13–24.

[Ji and Eisenstein2015] Yangfeng Ji and Jacob Eisenstein. 2015. One vector is not enough: Entity-augmented distributed semantics for discourse relations. *Transactions of the Association for Computational Linguistics*, 3:329–344.

[Ji et al.2016] Yangfeng Ji, Gholamreza Haffari, and Jacob Eisenstein. 2016. A latent variable recurrent neural network for discourse relation language models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Jones et al.2006] Rosie Jones, Benjamin Rey, Omid Madani, and Wiley Greiner. 2006. Generating query substitutions. In *Proceedings of the 15th international conference on World Wide Web*, pages 387–396. ACM.

[Jordan and Ng2002] Michael Jordan and Andrew Ng. 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14:841.

[Kalchbrenner and Blunsom2013] Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent convolutional neural networks for discourse compositionality. In *Proceedings of the Workshop on Continuous Vector Space Models and their Compositionality*, pages 119–126, Sofia, Bulgaria, August. Association for Computational Linguistics.

[Kiros et al.2015] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284.

[Klein and Manning2003] Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *the 41st Annual Meeting*, pages 423–430, Morristown, NJ, USA. Association for Computational Linguistics.

[Kohonen1990] Teuvo Kohonen. 1990. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.

[Krizhevsky et al.2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Lascarides and Asher2007] Alex Lascarides and Nicholas Asher. 2007. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer.

[Le and Mikolov2014] Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*.

[LeCun et al.2010] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. 2010. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE.

[Lee et al.2011] Heeyoung Lee, Yves Peirsman, Angel Chang, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2011. Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task*, pages 28–34. Association for Computational Linguistics.

[Lee et al.2013] Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu, and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. *Computational Linguistics*.

[Lee2001] Lillian Lee. 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Artificial Intelligence and Statistics*, volume 2001, pages 65–72.

[Levin1993] Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*, volume 348. University of Chicago press Chicago.

[Li and Nenkova2014] Junyi Jessy Li and Ani Nenkova. 2014. Reducing sparsity improves the recognition of implicit discourse relations. In *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, pages 199–207, Philadelphia, PA, U.S.A., June. Association for Computational Linguistics.

[Li et al.2015] Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015. When are tree structures necessary for deep learning of representations? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2304–2314, Lisbon, Portugal, September. Association for Computational Linguistics.

[Lin et al.2009] Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 343–351. Association for Computational Linguistics.

[Lin et al.2010a] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2010a. A PDTB-Styled End-to-End Discourse Parser. *arXiv.org*, November.

[Lin et al.2010b] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2010b. A pdtb-styled end-to-end discourse parser. *CoRR*, abs/1011.0835.

[Lin et al.2014] Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(02):151–184.

[Littlestone1988] Nick Littlestone. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318.

[Ljubesic et al.2008] N Ljubesic, Damir Boras, Nikola Bakaric, and Jasmina Njavro. 2008. Comparing measures of semantic similarity. In *Information Technology Interfaces, 2008. ITI 2008. 30th International Conference on*, pages 675–682. IEEE.

[lou2014] 2014. *Verbose, Laconic or Just Right: A Simple Computational Model of Content Appropriateness under Length Constraints*, Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics.

[Louis and Nenkova2010] Annie Louis and Ani Nenkova. 2010. Creating local coherence: An empirical assessment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 313–316. Association for Computational Linguistics.

[Louis et al.2010] Annie Louis, Aravind Joshi, Rashmi Prasad, and Ani Nenkova. 2010. Using entity features to classify implicit discourse relations. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 59–62. Association for Computational Linguistics.

[Mann and Thompson1988] William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

[Marchetti-Bowick and Chambers2012] Micol Marchetti-Bowick and Nathanael Chambers. 2012. Learning for microblogs with distant supervision: Political forecasting with twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 603–612. Association for Computational Linguistics.

[Marcu and Echihabi2002a] Daniel Marcu and Abdessamad Echihabi. 2002a. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 368–375. Association for Computational Linguistics.

[Marcu and Echihabi2002b] Daniel Marcu and Abdessamad Echihabi. 2002b. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 368–375. Association for Computational Linguistics.

[Marcu1999] Daniel Marcu. 1999. Discourse trees are good indicators of importance in text. *Advances in automatic text summarization*, pages 123–136.

[Marcus et al.1993] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

[McCallum2002] Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://www.cs.umass.edu/ mccallum/mallet.

[McClelland and Patterson2002] James L McClelland and Karalyn Patterson. 2002. Rules or connections in past-tense inflections: What does the evidence rule out? *Trends in cognitive sciences*, 6(11):465–472.

[Mikolov et al.2013a] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

[Mikolov et al.2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Mintz et al.2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

[Mohamed et al.2012] Abdel-rahman Mohamed, George E Dahl, and Geoffrey Hinton. 2012. Acoustic modeling using deep belief networks. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(1):14–22.

[Oza et al.2009] Umangi Oza, Rashmi Prasad, Sudheer Kolachina, Dipti Misra Sharma, and Aravind Joshi. 2009. The hindi discourse relation bank. In *Proceedings of the third linguistic annotation workshop*, pages 158–161. Association for Computational Linguistics.

[Park and Cardie2012] Joonsuk Park and Claire Cardie. 2012. Improving implicit discourse relation recognition through feature set optimization. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 108–112. Association for Computational Linguistics.

[Paşca et al.2006] Marius Paşca, Dekang Lin, Jeffrey Bigham, Andrei Lifchits, and Alpa Jain. 2006. Names and similarities on the web: fact extraction in the fast lane. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 809–816. Association for Computational Linguistics.

[Patterson and Kehler2013] Gary Patterson and Andrew Kehler. 2013. Predicting the presence of discourse connectives. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

[Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, 12:1532–1543.

[Petrov et al.2006] Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

[Pitler and Nenkova2008] Emily Pitler and Ani Nenkova. 2008. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195. Association for Computational Linguistics.

[Pitler et al.2008] Emily Pitler, Mridhula Raghupathy, Hena Mehta, Ani Nenkova, Alan Lee, and Aravind K Joshi. 2008. Easily identifiable discourse relations. *Technical Reports (CIS)*, page 884.

[Pitler et al.2009] Emily Pitler, Annie Louis, and Ani Nenkova. 2009. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 683–691. Association for Computational Linguistics.

[Polyak1964] Boris Teodorovich Polyak. 1964. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17.

[Prasad et al.2007] Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, Livio Robaldo, and Bonnie L Webber. 2007. The penn discourse treebank 2.0 annotation manual.

[Prasad et al.2008] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*. Citeseer.

[Raghunathan et al.2010] Karthik Raghunathan, Heeyoung Lee, Sudarshan Rangarajan, Nathanael Chambers, Mihai Surdeanu, Dan Jurafsky, and Christopher Manning. 2010. A multi-pass sieve for coreference resolution. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 492–501, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Ratinov et al.2011] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics.

[Reschke et al.2014] Kevin Reschke, Martin Jankowiak, Mihai Surdeanu, Christopher Manning, and Daniel Jurafsky. 2014. Event extraction using distant supervision. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

[Riedmiller2005] Martin Riedmiller. 2005. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *Machine Learning: ECML 2005*, pages 317–328. Springer.

[Riloff et al.1999] Ellen Riloff, Rosie Jones, et al. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479.

[Ritter et al.2011] Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.

[Rutherford and Xue2014] Attapol T. Rutherford and Nianwen Xue. 2014. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014)*, Gothenburg, Sweden, April.

[Rutherford and Xue2015] Attapol Rutherford and Nianwen Xue. 2015. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 799–808, Denver, Colorado, May–June. Association for Computational Linguistics.

[Schourup1999] Lawrence Schourup. 1999. Discourse markers. *Lingua*, 107(3):227–265.

[Senior et al.2015] Andrew Senior, Hasim Sak, Felix de Chaumont Quitry, Tara N. Sainath, and Kanishka Rao. 2015. Acoustic modelling with cd-ctc-smbr lstm rnns. In *ASRU*.

[Socher et al.2012] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.

[Socher et al.2013a] Richard Socher, John Bauer, Christopher D Manning, and Andrew Y Ng. 2013a. Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*. Citeseer.

[Socher et al.2013b] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

[Song et al.2015] Yangqiu Song, Haoruo Peng, Parisa Kordjamshidi, Mark Sammons, and Dan Roth. 2015. Improving a pipeline architecture for shallow discourse parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 78–83, Beijing, China, July. Association for Computational Linguistics.

[Speriosu et al.2011] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter polarity classification with label propagation over lexical links and the follower graph. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 53–63. Association for Computational Linguistics.

[Sporleder and Lascarides2008] Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(03):369–416.

[Stone et al.1968] Philip Stone, Dexter C Dunphy, Marshall S Smith, and DM Ogilvie. 1968. The general inquirer: A computer approach to content analysis. *Journal of Regional Science*, 8(1).

[Sundermeyer et al.2012a] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012a. Lstm neural networks for language modeling. In *INTERSPEECH*.

[Sundermeyer et al.2012b] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. 2012b. Lstm neural networks for language modeling. In *INTERSPEECH*.

[Tai et al.2015] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

[Thamrongrattanarit et al.2013] Attapol Thamrongrattanarit, Colin Pollock, Benjamin Goldenberg, and Jason Fennell. 2013. A distant supervision approach for identifying perspectives in unstructured user-generated text. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 922–926, Nagoya, Japan, October. Asian Federation of Natural Language Processing.

[Toutanova and Manning2000] Kristina Toutanova and Christopher D Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics.

[Toutanova et al.2003] Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

[Turian et al.2010] Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics.

[Vinyals et al.2015] Oriol Vinyals, Ł ukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015. Grammar as a foreign language. In C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2755–2763. Curran Associates, Inc.

[Wang and Lan2015] Jianxiang Wang and Man Lan. 2015. A refined end-to-end discourse parser. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 17–24, Beijing, China, July. Association for Computational Linguistics.

[Wang et al.2012] Xun Wang, Sujian Li, Jiwei Li, and Wenjie Li. 2012. Implicit discourse relation recognition by selecting typical training examples. In *Proceedings of COLING 2012*, pages 2757–2772, Mumbai, India, December. The COLING 2012 Organizing Committee.

[Wang et al.2015] Longyue Wang, Chris Hokamp, Tsuyoshi Okita, Xiaojun Zhang, and Qun Liu. 2015. The dcu discourse parser for connective, argument identification and explicit sense classification. *CoNLL 2015*, page 89.

[Xue et al.2015] Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Rashmi Prasad, Christopher Bryant, and Attapol Rutherford. 2015. The conll-2015 shared task on shallow discourse parsing. In *Proceedings of the Nineteenth Conference on Computational Natural Language Learning - Shared Task*, pages 1–16, Beijing, China, July. Association for Computational Linguistics.

[Yao et al.2010] Limin Yao, Sebastian Riedel, and Andrew McCallum. 2010. Collective cross-document relation extraction without labelled data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023. Association for Computational Linguistics.

[Yoshida et al.2015] Yasuhisa Yoshida, Katsuhiko Hayashi, Tsutomu Hirao, and Masaaki Nagata. 2015. Hybrid approach to pdtb-styled discourse parsing for conll-2015. *CoNLL 2015*, page 95.

[Zeiler2012] Matthew D Zeiler. 2012. Adadelta: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

[Zeyrek and Webber2008] Deniz Zeyrek and Bonnie L Webber. 2008. A discourse resource for turkish: Annotating discourse connectives in the metu corpus. In *IJCNLP*, pages 65–72.

[Zhou and Xue2012] Yuping Zhou and Nianwen Xue. 2012. Pdtb-style discourse annotation of chinese text. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 69–77. Association for Computational Linguistics.

[Zhou and Xue2015] Yuping Zhou and Nianwen Xue. 2015. The chinese discourse treebank: A chinese corpus annotated with discourse relations. *Language Resources and Evaluation*, 49(2):397–431.