Improve Classification on Infrequent Discourse Relations via Training Data Enrichment

by

Kailang Jiang

B. Eng., Shanghai Jiao Tong University, 2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Computer Science)

The University of British Columbia
(Vancouver)

November 2016

Abstract

Discourse parsing is a popular technique widely used in text understanding, sentiment analysis, and other NLP tasks. However, for most discourse parsers, the performance varies significantly across different discourse relations. In this thesis, we first validate the underfitting hypothesis, i.e., the less frequent a relation is in the training data, the poorer the performance on that relation. We then explore how to increase the number of positive training instances, without resorting to manually creating additional labeled data. We propose a training data enrichment framework that relies on co-training of two different discourse parsers on unlabeled documents. Importantly, we show that co-training alone is not sufficient. The framework requires a filtering step to ensure that only "good quality" unlabeled documents can be used for enrichment and re-training. We propose and evaluate two ways to perform the filtering. The first is to use an agreement score between the two parsers. The second is to use only the confidence score of the faster parser. Our empirical results show that agreement score can help to boost the performance on infrequent relations, and that the confidence score is a viable approximation of the agreement score for infrequent relations.

Preface

This dissertation is an original intellectual product of the author, Kailang Jiang. The author conducted all the experiments and wrote the manuscript, under the supervision of Dr. Giuseppe Carenini and Dr. Raymond Ng.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to offer my wholehearted gratitude to everyone who has inspired or supported my work during my master study.

Special thanks to my supervisors Dr. Giuseppe Carenini, and Dr. Raymond Ng! Thanks you for teaching me how to do research and how to write a paper, providing me so many good ideas and suggestions during every meeting, giving me very detailed feedback on everything, and always being so patient and encouraging.

Thank all my fellow students and friends in UBC for providing me your useful experiences and suggestions on study, research and life, and for all your concern, company and support.

Particular thanks to my good friends and my parents that are far away. Your unconditional love and support are the reason why I want to become a better person.

Chapter 1

Introduction

"Clauses and sentences rarely stand on their own in an actual discourse; rather, the relationship between them carries important information that allows the discourse to express a meaning as a whole beyond the sum of its individual parts. Discourse analysis seeks to uncover this coherence structure." (Joty, et al., 2015) [16]

Research and application of natural language processing (NLP) has been growing rapidly in the past decade, and the value of discourse structure and relation in NLP is getting more and more attention. Discourse parsing, which discovers how sentences and clauses are connected together, is now widely used in many NLP tasks, including text understanding [2], machine translation evaluation [11], sentiment analysis [3], text summarization [10], etc.

Studies in the past decade on discourse parsing, such as [3], [16], have greatly improved the performance of discourse parsing in general. However, it has been observed that the performance across the discourse relations varies significantly [3], and that poor performance may be linked to underfitting, i.e., a lack of training data [16]. In this thesis, we investigate the underfitting hypothesis and study how to improve the situation.

1.1 Discourse Parsing

Most text analysis tasks focus on only properties of each single sentence or clause. However, those sentences and clauses are not arbitrarily put together, the way they are constructed and related in fact carries a lot of important information that could not be discovered from their segregated parts. And capturing the relations between sentences and clauses can help us better understand the entire text.

Consider the following two examples:

- *"While the pound has attempted to stabilize, currency analysts say it is in critical condition."*
- *"The pound has attempted to stabilize, currency analysts say it is in critical condition."*

The first sentence is extracted from an Wall Street Journal article in our unlabeled dataset that will be further introduced in Section 4.1. And the second sentence just takes away the first conjunction "While" from the first sentence, which does not impair the understanding of the first clause. Most readers will find the first sentence easy to understand, while the second sentence confusing, since the relation between pound's stabilization and critical condition is not clear. Normally authors always try to construct their text in a coherent and logical way so that it is easy to interpret and understand. So uncovering the the coherence structure underneath the text is very helpful for understanding it, and is the foundation of discourse analysis.

A multi-sentential discourse parser takes a document as input, and returns its discourse structure that shows how clauses and sentences are related in the document, via the use of various discourse relations. For instance, the very popular corpus - Rhetorical Structure Theory Discourse Treebank (RST-DT) [5] groups different types of discourse relations between sentences and clauses into 18 classes, as listed in Table 1.1.

As an illustration, Figure 1.1 shows the discourse structure of the first sentence "While the pound has attempted to stabilize, currency analysts say it is in critical condition." produced by CODRA[16] - one of the state-of-the-art discourse parsers used in this thesis. As shown in the figure, this discourse tree has three

Elaboration	Joint	Attribution
Same-Unit	Contrast	Explanation
Background	Cause	Temporal
Enablement	Comparison	Evaluation
Topic-Comment	Condition	TextualOrganization
Topic-Change	Manner-Means	Summary

Table 1.1: 18 Discourse Relations in RST-DT Dataset

leaves that correspond to contiguous atomic text spans, called elementary discourse units (EDUs). EDUs are clause-like units that serve as building blocks [38]. Adjacent EDUs are then related by coherence relations (e.g., Attribution, Contrast), thereby forming larger units (represented by internal nodes), which in turn are also linked by coherence relations. Discourse units linked by a relation are further distinguished based on their relative importance in the text: the nucleus being the central part, whereas satellites are peripheral ones. For example, in Figure 1.1, "Attribution" is a relation between a nucleus (EDU 3) and a satellite (EDU 2), and "Contrast" is a relation between a nucleus (EDU [2, 3]) and a satellite (EDU 1). From this discourse tree we can see clearly that the clause "While the pound has attempted to stabilize" contrasts with the clause "currency analysts say it is in critical condition".

A better understanding of such relations between clauses and sentences is very helpful to many NLP tasks such as text understanding [2], machine translation evaluation [11], sentiment analysis [9] [3], text summarization [10], etc. For instance, when we try to analyze the sentiment of the sentence mentioned above - "While the pound has attempted to stabilize, currency analysts say it is in critical condition.", if we only look at the properties of each word without considering the discourse structure, we might regard the word "stabilize" to be positive and the word "critical" to be negative, and when we sum it up, it will be hard to tell what type of sentiment this sentence is carrying in total. But if we incorporate its discourse structure and relations shown in Figure 1.1, we will find that the two clauses contrast with each other and the second half is the nucleus. So the negative word in the second clause should be given a higher weight when calculating the sentiment

of the entire sentences, and get a more accurate result.

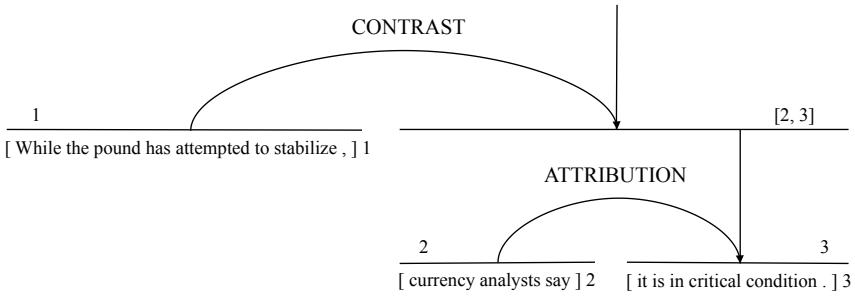


Figure 1.1: Discourse Tree of the Example Sentence

1.2 Motivation

Since a better understanding of discourse structures and relations can bring great benefits to so many NLP tasks, a lot of work has been done recently to improve the performance of discourse parsing [15] [8] [16]. However, all these works suffer from the same problem of lacking training data for certain discourse relations, which prevents them from achieving a better result on those relations.

It has been observed that the performance across the discourse relations varies significantly[3]. Meanwhile, different discourse relations are usually unevenly distributed in a dataset, and some of them occur much less frequently than other relations. We call the former the *infrequent* relations. For example, in the RST-DT corpus [5] which contains 385 documents, the frequency of "Elaboration" is 31.04%, while the frequency of "Summary" is only 0.88%.

In another benchmark corpus the Penn Discourse Treebank (PDTB) 2.0 [25], which contains about 2400 documents with discourse relations labeled for each pair of adjacent sentences, the relation "Conjunction" occurs 8759 times through the entire corpus, while the relations "Exception" and "Pragmatic concession" only appear 17 and 12 times respectively [12].

It has been noticed that poor performance of one discourse relation may be linked to **underfitting**, i.e., a lack of training data [16]. And given that the perfor-

mance of most discourse parsers depends on the availability of training data, the key question here is whether underfitting affects the infrequent relations more than the frequent ones. In Section 4.2, we will explicitly show that parsing performance of relations is correlated with the frequency of the relations.

Clearly, every discourse relation, infrequent or not, would benefit from the availability of more high-quality training data. However, creating such high-quality labeled data takes much time and effort to manually annotate documents with their discourse structures and relations. The question here is whether the infrequent relations are worthy of the extra effort required. It turns out that many infrequent relations actually play important roles in various NLP tasks. For example, the "Comparison" relation from RST-DT is known to indicate disagreement in a conversation [14] [2]. Moreover, the "Instantiation" relation from PDTB is regarded as an important feature for sentence specificity prediction [19]. Since these infrequent relations are very important to many NLP tasks as mentioned above, it is clearly worth extra efforts to acquire more training instances for infrequent relations.

1.3 Approach and Contributions

The main objective of this thesis is to explore how to mitigate the underfitting problem for infrequent relations - without manually creating labeled data for those relations. In particular, we aim to exploit the availability of a much larger amount of unlabeled data, with the help of a small amount of existing labeled data. That is, we need to adopt a semi-supervised learning algorithm and apply it on our discourse parsing problem here.

So the first step of our approach is to apply existing discourse parsers to the unlabeled data to generate more instances of infrequent relations, which are then used to re-train the existing parsers. Such co-training approaches have proved to be effective in solving similar problems in natural language processing [19] and information retrieval [4].

There is, however, a fatal flaw relying on co-training alone. If existing discourse parsers are poor in determining infrequent relations, the extra (re-)training instances of infrequent relations created from unlabeled data may not be of high quality. Indeed, adding poor quality re-training instances would exacerbate the un-

derfitting problem of infrequent relations. The second step of our approach is to apply a *filtering* step to the instances created from unlabeled data. The intention behind the filtering step is to *enrich* the re-training - that is, to select only the "high quality" instances to be used for re-training.

The workflow of our enrichment approach will be further described in Section 3.1, when it is applied to two discourse parsers, P1 and P2. The two parsers are initially trained on labelled data and then are applied to unlabelled data to generate new high-quality Training Examples for further re-training.

Our experiments have shown that the performance one a discourse relation is releted to its frequency in the dataset, and our empirical results show that agreement score filtering can boost the performance of infrequent relations considerably, and the confidence score of the SR-parser can also be used as a fast approximation of the agreement score. So far our results show that our data enrichment framework is not effective for frequent relations, but filtering on the node level with different threshold for infrequent and frequent relations has shown to be helpful to improve this situation. We believe that with more unlabeled documents and a more precise sliding threshold for every relation, the performance can continue to improve.

The specific contributions of our thesis are as follow:

- We explore one form of enrichment based on the notion of *agreement score* between two discourse parsers. Inspired by the theory on the success of ensembling for general classification [6], we choose two very different discourse parsers, namely the CKY-like CODRA parser by [16] and the Shift-Reduce (SR) parser by [15]. While Section 3.2 will give more details on why these two parsers are chosen, the key is that the parsers are based on very different algorithms and feature sets for discourse parsing. Our agreement score is based on the F-score measure for comparing discourse trees as proposed in [22]. Only the discourse relation instances in discourse trees with high-enough agreement scores pass through the filter for re-training purposes. Chapter 4 will show that such enrichment with agreement score improves the performance of infrequent relations.

- We explore another form of enrichment based on just the confidence score of the SR-parser. The rationale is that while the CODRA parser is generally

more accurate than the SR-parser, the SR-parser is two orders of magnitude faster. If a high-enough threshold on the confidence score of the SR-parser is used for enrichment, Chapter 4 will investigate whether the confidence score is a good approximation of the agreement score. If this approach is successful, an even larger number of unlabeled documents can be parsed rapidly to be used for re-training.

The advantages of our approach include:

- It is effective to boost the performance of infrequent relations.
- It makes good use of unlabeled documents and does not require any extra manual labeling.
- It takes advantage of two very different discourse parsers, and combines them together to reach better performance.
- The confidence score of the SR-parser is a good approximation of agreement score for filtering, so a larger number of unlabeled documents can be parsed rapidly to be used for re-training.

Disadvantages of our approach can be:

- It is not very effective on frequent discourse relations so far, while they consist of most of the testing instances, the total performance does not have a significant boost.
- It might require the two discourse parsers used in the co-training algorithm to be very different in order to achieve a good result, thus in the future if better discourse parsers appear we can not just take any two of them to feed into our framework, and results can be parser specific.

1.4 Outline

In Chapter 2, we will introduce existing discourse parsers, both those earlier discourse parsers and state-of-the-art discourse parsers, including the two parsers used in our framework. We will also introduce how other researchers have tried to enrich

training examples for discourse parsing, and provide more background information and applications of co-training algorithm. In Chapter 3, we will describe the details of our approach, including the workflow and enrichment process, the choices of the two parsers, filtering at different granularity, and different ways to set the threshold. Chapter 4 will show various experiments we have performed and the results and analysis. And in the end, Chapter 5 will summarize the contributions of this thesis and discuss future work.

Chapter 2

Related Work

In this chapter, we discuss some related work that has inspired our approach, or provided us the tools and information we needed to conduct our experiments. Section 2.1 introduces several existing discourse parsers, including the two that will be used in our experiments. Section 2.2 explores how other researchers tried to tackle the training data sparsity problem in discourse parsing, both for implicit relation classification and infrequent relation classification. Section 2.3 provides a brief description of the Co-training algorithm and its application in natural language processing and related areas.

2.1 Existing Discourse Parsers

In the early stage of discourse parsing research, (Marcu, 1999) [21] used machine learning techniques to build a shift-reduce discourse parser, which relies on decision tree classifiers to learn the rules from training data. To learn the shift-reduce actions, the discourse parser encodes five types of features: lexical (e.g, discourse cues), shallow-syntactic, similarity, operational (previous n shift-reduce operations) and discourse sub-structural features. Though its performance is not comparable to recent parsers, this work has inspired many recent machine learning approaches in discourse parsing.

In 2003, (Soricut et al., 2003) [27] developed the SPADE system that comes with probabilistic models for sentence-level discourse parsing. Their segmentation

and parsing models are based on lexicosyntactic patterns (features) extracted from the lexicalized syntactic tree of a sentence. The discourse parser uses an optimal parsing algorithm to find the most probable rhetorical tree structure for a sentence. SPADE was trained and tested on the RST-DT corpus. This work, by showing empirically the connection between syntax and discourse at the sentence level, has greatly influenced all major contributions in this area ever since. However, it is limited in several ways. First, SPADE does not produce a full-text (document-level) parse. Second, its parsing model makes an independence assumption between the label and the structure of a discourse tree constituent, and it ignores the sequential and the hierarchical dependencies between the constituents. Third, it relies only on lexico-syntactic features, and it follows a generative approach to estimate the model parameters.

In 2010, (Hernault et al., 2010) [13] introduced the HILDA system that is based on Support Vector Machines (SVMs). It feeds the lexical and syntactic features used in SPADE plus more context to its segmenter, which is a binary SVM classifier. While for the discourse parser, SVM classifiers are applied iteratively, two at a time, one used to decide which adjacent unit to merge, the other used to choose the most reasonable relation label between the selected units. They report improved performance in discourse parsing on the RST-DT corpus.

On the other hand, (Subba et al., 2009) [29] proposes a shift-reduce parser that uses Inductive Logic Programming (ILP) to learn first-order logic rules from a large set of features for relation labeling, including the rich compositional semantics from a semantic parser. This work shows that compositional semantics with other features are helpful to improve relation classification performance.

However, both HILDA and the ILP-based approach mentioned above have several limitations. First, they do not differentiate between intra-sentential parsing and multi-sentential parsing, and use a single uniform model in both scenarios. Second, they take a greedy (sub-optimal) approach to construct a discourse tree. Third, they disregard sequential dependencies between discourse tree constituents, which has been recently shown to be critical by [7]. Furthermore, HILDA considers the structure and the labels of a discourse tree separately.

Recent works [16] [15] have overcome these constraints and improved the performance and efficiency of discourse parser. (Joty et al.) [17] [16] proposed

a Cocke-Kasami-Younger(CKY)-like discourse parser which tries to build a discourse tree by applying an optimal parsing algorithm to the probabilities of all the constituents inferred from two conditional random fields (CRFs) jointly: a linear chain dynamic-CRF for intra-sentential parsing, and a uni-variate graphical model for multi-sentential parsing. It combines the results returned by the two parsers to build the final discourse tree. A log of features are used to improve the classifier, including ngrams, lexical chains, dominance set, contextual and sub-structure features, etc. The application of CRFs to discourse parsing problem has showed to improve the parsing performance at both intra and multi sentential level. However, its inefficiency in terms of both speed and space makes it impractical in large applications.

Based on Joty's idea [17], (Feng et al., 2014) [8] proposed a linear time discourse parser. They made several modifications in order to reduce the complexity: greedy bottom-up parsing procedure that allows linear time parsing; usage of the linear chain CRF in both intra and multi-sentential parsing; separated modeling of structure and relation; novel idea of post-editing which does a second pass parsing to incorporate information from upper-level discourse constituents, etc. They also adopted additional features that are not used in [17], which help them achieve better accuracy.

On the other hand, [15] proposed a representation learning approach for discourse parsing which formalizes discourse tree building process as a sequence of decision problems by using a transition-based shift-reduce parser. It jointly learns a linear transformation from unigrams to lower dimensional latent space representation and a SVM decision classifier in this space to make shift-reduce decisions. We have reproduced the result of this parser on the RST-DT dataset, and the result shows that it does have a great advantage over [17] and [8] in terms of efficiency, while it is the other way around concerning the performance.

Furthermore, these existing parsers all suffer from the same problem of training data sparsity, as it takes too much time and effort to manually annotate documents with their discourse structures and relations. So in the next section, we will investigate existing works in enriching training data for discourse parsing.

2.2 Training Data Expansion in Discourse Parsing

The training data sparsity problem impacts several aspects of discourse parsing. In this section, we first introduce the one for parsing implicit relations. Experiences in expanding training data for implicit relations have inspired us to tackle the second problem, training data enrichment for infrequent relations — the key issue in this thesis.

2.2.1 Training Data Expansion for Implicit Relations

A key distinction in discourse parsing is between explicit and implicit relations. The former are signaled by a cue phrase like "because" while the latter are not and consequentially are more difficult to identify. Several studies have been conducted to tackle the problem of classifying implicit relations which do not have many explicit features and examples. (Zhou et al., 2010) [31] presents a method to predict the missing connective based on a language model trained on an unlabeled corpus. The predicted connective is then used as a feature to classify the implicit relation. (Mckeown et al., 2013) [24] tackles the feature sparsity problem by aggregating implicit relations into larger groups. (Lan et al., 2013) [18] combines different data through multi-task learning. The method performs implicit and explicit relation classification in PDTB framework as two tasks and relies on multi-task learning to obtain higher performance.

[20] proposes a multi-task neural networks that combines RST-DT, PDTB and unlabeled data together through multi-task learning process, and gets performance improvements on implicit relations, though they only apply their scheme on the four coarse top-level relation types. Their scheme is based on retrieving more training instances from unlabeled data through cue phrases. This approach of using explicit examples to predict implicit examples has been shown to produce mixed results [28]. Moreover, [16] has shown that there are many more features beyond cue phrases that are useful for discourse parsing.

Though training data expansion for implicit relations are different from that for infrequent relations, we can still get a lot of insights from it about what may or may not be effective in producing more useful training data.

2.2.2 Training Data Expansion for Infrequent Relations

[12] proposes a feature vector extension approach to improve classification of infrequent discourse relations. The approach is based on word co-occurrence. They propose the method that first computes the co-occurrence between features using unlabeled data and use that information to extend the feature vectors during training and testing, thereby reducing the sparseness in test feature vectors. Partly because a simple discourse parser was used, their approach is shown to produce only minimal improvements in performance.

Unlike [20] and [12], we aim to exploit more advanced parsers with higher performance, and also keep the finer-granularity of the relations, especially focusing on the infrequent relations.

2.3 Co-training

Co-training is a semi-supervised learning technique first introduced by [4], with its application in helping the search engine better classify whether a webpage is an "academic course home page". It requires two views of the data and assumes that each example is described using two different feature sets that provide different, complementary information about the instance. Ideally, the two views are conditionally independent (i.e., the two feature sets of each instance are conditionally independent given the class) and each view is sufficient (i.e., the class of an instance can be accurately predicted from each view alone). Co-training first learns a separate classifier for each view using any labeled examples. The most confident predictions of each classifier on the unlabeled data are then used to iteratively construct additional labeled training data. After one thousand iterations, the classifier reaches very high accuracy with a very small amount of initial labeled web pages as training examples.

Similar co-training efforts have been found to be effective in many NLP problems when only a small amount of labeled data is available. For example, [30] proposes a co-training approach for cross-lingual sentiment classification, which leverages an available English corpus for Chinese sentiment classification by using the English corpus as training data. Machine translation services are used for eliminating the language gap between the training set and test set, and English features

and Chinese features are considered as two independent views of the classification problem.

While [19] applies co-training on predicting sentence specificity. To train their semi-supervised model for sentence specificity, they use a repurposed corpus of binary annotations of specific and general sentences drawn from Wall Street Journal articles originally annotated for discourse analysis, and then make use of unlabeled data from New York Times and Wall Street Journal articles (no overlap between them and the labeled examples and the testing data) for co-training.

However, there's a fatal flaw relying on co-training alone, as we have previously discussed in Section 1.3. If existing discourse parsers are poor in determining infrequent relations, the extra (re-)training instances of infrequent relations created from unlabeled data may not be of high quality, and might exacerbate the underfitting problem of infrequent relations. So in the next section, we will describe our approach that adopts the idea of co-training algorithm with a filtering step, and combine the advantages of recent discourse parsers to select the "high quality" instances for re-training. And this is what we mean by **enrichment** — different from simply expanding the training set with more data, we also control the quality of new training instances through filtering out the unconfident ones.

Chapter 3

Enrichment Approach

3.1 Workflow

The general workflow of our enrichment approach is shown in Figure 3.1, when it is applied to two discourse parsers, P1 and P2. First we use the labeled data to provide initial training of the two parsers. Then each parser is used to produce a discourse tree for each unlabeled document. After that, we apply a filtering step to select those "high quality" discourse trees, which are added to the original labeled data to form the "enriched training data" to re-train the two parsers.

3.2 Selection of Discourse Parsers

In our approach, the first parser we pick is the CODRA parser [16], which applies a CKY parsing algorithm to probabilities inferred from two Conditional Random Fields for both intra-sentential and multi-sentential parsing. We pick the CODRA parser because of its optimal CKY parsing algorithm and its accuracy. The second parser we pick is the SR-parser [15], which transforms the surface features into a latent space that facilitates RST discourse parsing. The main advantage of the SR-parser is that it can train and parse documents in almost linear time (regarding the document length), while the CODRA parser needs cubic time. Our choice of the two parsers is partly based on the fact that they rely on very different algorithms and feature sets, which is desired by the co-training algorithm. Although another

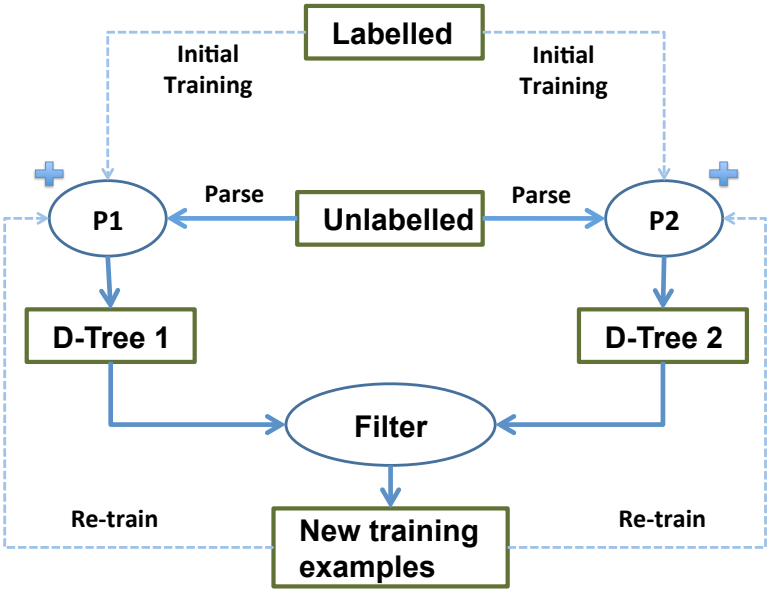


Figure 3.1: Workflow of Our Enrichment Approach

discourse parser [8] also delivers state-of-the-art performance, its approach and features are very similar to CODRA's, so we only wanted to select one of them. And due to the fact that Feng's parser is not publicly available and our existing experience on CODRA, we picked CODRA in our approach. Another reason of our choice on the SR-parser is that discourse parsing of documents in general can be slow in both training and parsing. Thus, the SR-parser is attractive in allowing us to explore the tradeoffs between accuracy and efficiency.

3.3 Enrichment Process

A co-training algorithm alone is not sufficient for the enrichment process, since both the CODRA parser and the SR-parser perform poorly for infrequent relations. The extra (re-)training instances of infrequent relations created from unlabeled data may not be of high quality. The key idea is to enrich the re-training by selecting only the **"high quality"** instances. In this thesis we investigate two forms of enrichment, based on the agreement score between the two parsers, and the confidence

score given by each parser individually.

To produce the agreement score between the two parsers, we use both parsers to parse every unlabeled document. Then we treat the parse tree produced by the CODRA parser as the ground truth, and the one produced by the SR-parser as testing, and use the F-score for comparing discourse trees proposed in [22] as the **agreement score**. Finally, if the agreement score passes a preset threshold, the unlabeled document is regarded as reliable and the discourse tree is added to enrich re-training.

The second form of enrichment examined in this thesis is based on using the **confidence score** of each parser individually. Instead of using both parsers to parse the same document and compute the agreement score, we use the confidence score given by only one parser when it produces a discourse tree for one document as the criteria to filter new discourse trees added to re-train this discourse parser. The advantage of using the confidence score produced by only one parser as an approximation of the agreement score between the two parsers is to reduce the amount of discourse parsing needed to produce and select new training instances, especially when one of the parser is much faster than the other one.

The SR-parser does not provide a confidence score for a discourse tree generated for a document directly. It generates a discourse tree by performing a set of actions. More specifically, each action creates a node in the tree by combining two text spans and by selecting a discourse relation for the pair. Since each action is chosen with a certain confidence score (which technically is the distance between the chosen action and the hyperplane, provided by the underlying Linear SVC algorithm), we use the average confidence of the actions performed to create the tree as the confidence score of the entire tree. If this approach is successful, an even larger number of unlabeled documents can be parsed rapidly for re-training.

As for the CODRA parser, it provides confidence score both for a relation label at one node and for the structure of an entire discourse tree. This give us more flexibility to choose which score to use and at which level to filter the data. Similar to the SR-parser, we can use the confidence score for the entire tree to filter new training instances at the document. Also, if we look at the confidence score for the label at one node, we can try to filter new training examples at a finer granularity as described in the next section.

3.4 Filtering at Finer Granularity

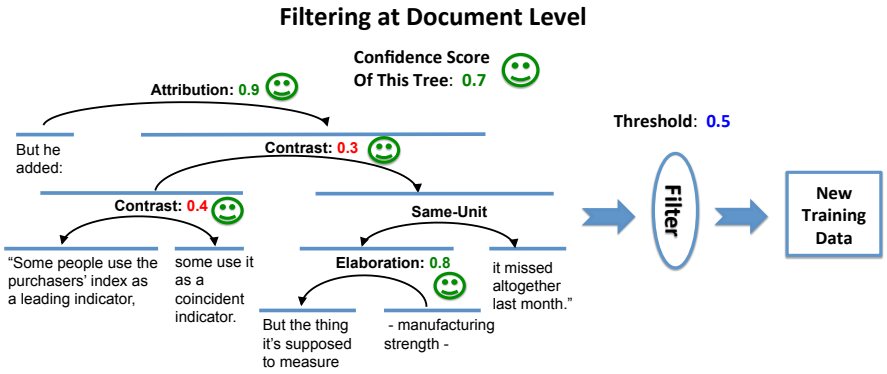
The filtering process described above is performed at the document level. That is, we calculate an agreement/confidence score for an entire discourse tree of a document, and if the score passes the threshold, this entire discourse tree along with every node on it will be added to the new training instances, even though some nodes on this tree may have low scores. In this case, these nodes with low scores are very likely to harm the performance of the discourse parser when added to its new training instances.

In order to reduce such noise brought by the low-score nodes in a high-score tree, we seek to filter at a finer granularity — the node level. That is, we compute the confidence/agreement score for each specific relation label of a node, and compare it to the threshold. If the confidence score of one node passes the threshold, this node will be added to the new training set, otherwise the node will be discarded, no matter if the entire discourse tree has a high score.

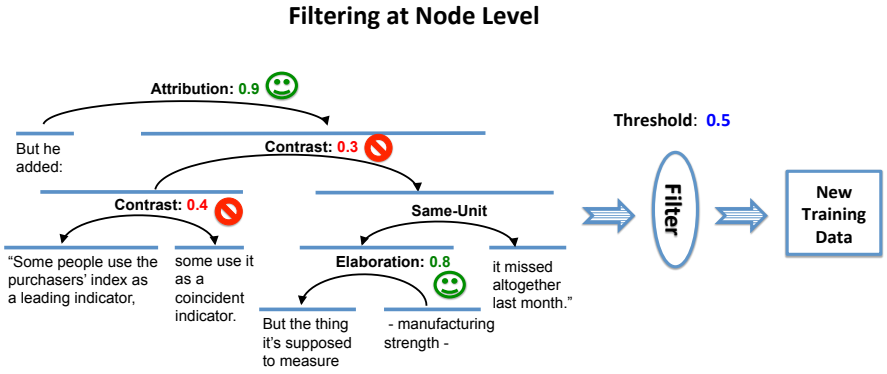
For example, as shown below in Figure 3.2, the threshold is set to 0.5 for both cases. In Figure 3.2(a), the confidence score of the discourse tree passes the threshold, every node on this tree will be added to the new training instances, even though the confidence scores of some nodes are below the threshold. While in Figure 3.2(b), each node's confidence score is compared to the threshold, and only nodes with confidence score higher than the threshold will be added to new training instances. So from the same discourse tree, the nodes we select to add to new training instances can be different under different filtering granularity.

In our approach, we will use both types of filtering under different situations. The advantage of filtering at a finer granularity is obvious, this way we can pick the high quality training instances more precisely, avoiding some "noisy nodes" that hide within a high quality discourse trees. However, it is not always possible to break a discourse tree and add only some of its unconnected pieces for re-training. For examples, the SR-parser will need the entire tree structure to do the training for document-level parsing. So we will apply node level filtering for the CODRA parser and document level filtering for the SR-parser.

Since node level filtering is possible for CODRA, we can have a more precise way to control the threshold. That is, we can set different thresholds for different



(a) At Document Level



(b) At Node Level

Figure 3.2: Filter at Different Granularity

types of nodes. More specifically, when the parser labels a node with one relation, depending on what relation it is, we can use different thresholds to determine whether this node can be added to the new training instances. More discussion about why we want to set different thresholds for different relations and how to set it can be found in Section 4.7. But generally, we need to be more strict with adding new training instances of frequent relations, while less strict with adding those of infrequent relations. A simple approach is to divide all the relations into two

groups, frequent relations and infrequent relations, according to their frequency in the gold standard dataset. Then we can use one threshold for the frequent relations, and a different threshold for those infrequent relations. If more precise control of the threshold is desired, a sliding threshold for every different relation can also be applied to node-level filtering.

Chapter 4

Empirical Evaluation

4.1 Datasets

In this thesis, we use the RST-DT dataset as the gold standard labeled data. It consists of 385 documents selected from Penn Treebank [23], which are all originally articles from the Wall Street Journal. Those 385 documents in the RST-DT dataset are divided into two fixed groups: the training set consisting of 347 documents, and the test set 38 documents. For results reported in this thesis, we used those 347 documents as the initial training set. The remaining 38 documents made up the test set used to evaluate the performance of the parser, which is re-trained using the enriched dataset.

For the unlabeled documents, we used 2000 Wall Street Journal articles from the Penn Treebank dataset [23]. In other words, the gold standard dataset and unlabeled dataset are from the same source; but there is no document belonging to both.

In discourse parsing, there are various performance measurements, such as on the structure (i.e., hierarchical spans) and the labels (i.e., nuclearity and relation classification). The results reported here focuses on relation classification. To evaluate the parsing performance based on the gold standard, we use the standard F-score measure, which is the harmonic mean of precision and recall [1]. More specifically, we use the F-score measure for comparing discourse trees, as proposed in [22].

4.2 The Underfitting Hypothesis: Performance vs Frequency

As for the discourse relations, we examine all the 18 coarse-grained relations introduced in Section 1.1. Figure 4.1 shows the most frequent and the least frequent five relations in all the 385 documents in the RST-DT dataset. We can see that the most frequent relations can be two order of magnitude higher in frequencies than those of the infrequent ones. For example, the "Elaboration" relation makes up over 31% of all the nodes in the entire dataset, while the "Topic Change" relation accounts for less than 0.5%.

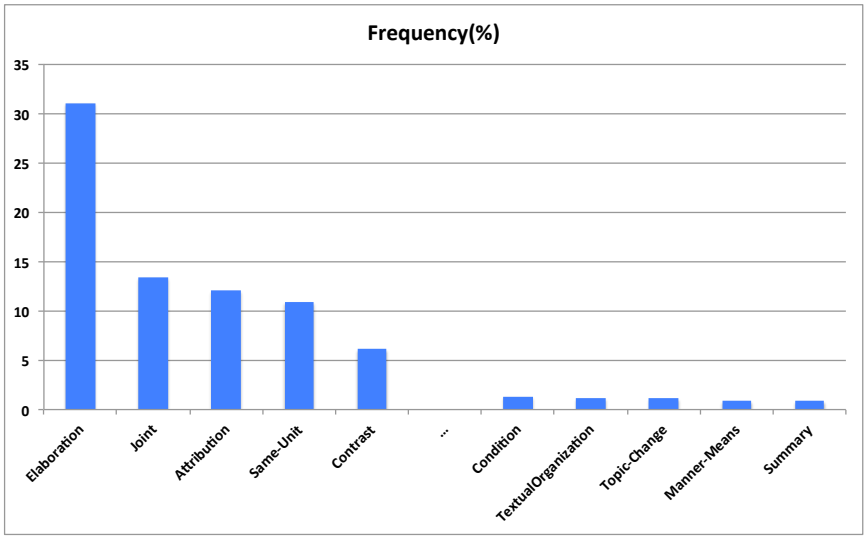


Figure 4.1: Distribution of the Most Frequent and the Least Frequent 5 Relations in RST-DT

Given the large disparity in relation frequencies, we next examine whether infrequent relations suffer from worse performance than the frequent relations, i.e., the underfitting hypothesis of a lack in training data of the infrequent relations. Here we used the 347 documents to train the SR-parser, and then tested the parser on the 38 documents. Figure 4.2 shows the performance of each relation (i.e., F-score) versus its frequency. We can see that for each relation, its performance has high correlation with its frequency. Indeed, the Pearson correlation coefficient is 0.87, validating the underfitting hypothesis. This suggests that it would be a

reasonable approach to boost the performance of infrequent relations by enriching their training instances.

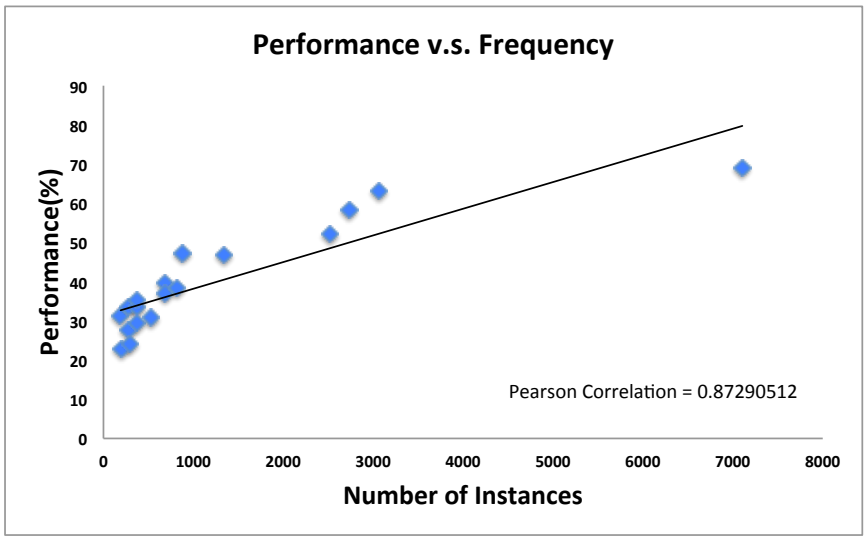


Figure 4.2: Performance versus Frequency for Each Relation

4.3 Effect of Enrichment on Infrequent Relations

The first form of enrichment examined below is based on the agreement score between the two parsers, as discussed in the previous section. Table 4.1 below shows the improvements on the F-scores from the SR-parser of the top-8 infrequent relations, based on a threshold of 0.5 in the filtering step. The different columns of the table show an increasing number of unlabeled documents used in enrichment, from 500 documents to 2000 documents. Figure 4.3 shows the relative F-score improvements across all the 18 relations, ranked from left to right in ascending order of frequency. As a specific example, the F-score of "Topic Change" improves 5.88% with 500 documents, and 13.15% with 2,000 documents.

As shown in the table and the figure, there is a positive effect on performance by enrichment based on the agreement score. The larger the number of unlabeled documents used, the higher is the gain in performance for the top-8 infrequent relations. The exact magnitude of the gain varies.

Relation	500	1000	1500	2000
Summary	2.13	2.80	3.91	5.16
Manner-Means	16.62	21.13	21.61	22.08
Topic-Change	5.88	7.21	12.88	13.15
TextualOrganization	1.42	3.31	7.49	8.14
Condition	3.91	8.69	12.44	18.55
Comparison	3.19	6.06	6.95	10.42
Evaluation	2.83	4.76	8.09	10.98
Topic-Comment	2.69	4.55	6.73	9.48

Table 4.1: Relative F-scores Improvements (%) on the Top-8 Infrequent Relations

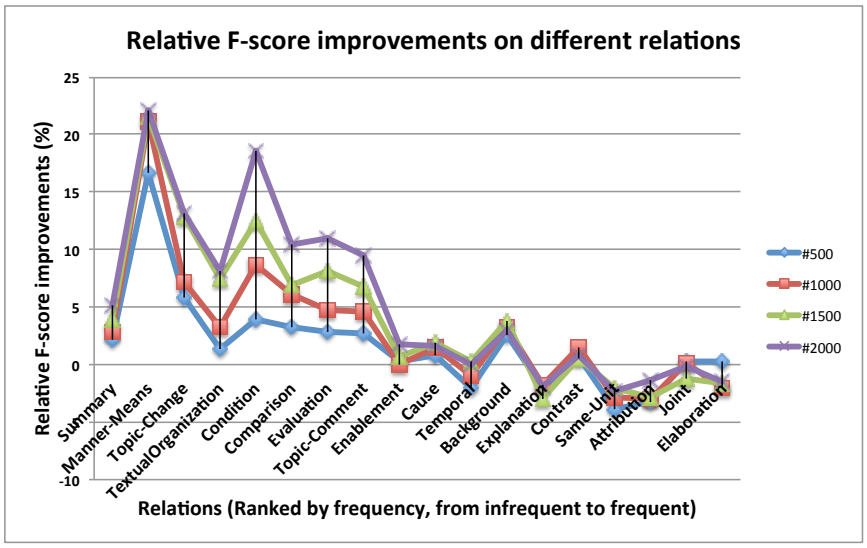


Figure 4.3: Relative F-score Improvements on Different Relations

So far we have described data enrichment in terms of the number of unlabeled documents. The more detailed analysis is to examine the actual number of training instances created from the unlabeled documents for each relation. Figure 4.4 shows the actual number of training instances added for each relation, represented as a percentage relative to the frequency of the instances in the original training dataset. For example, for the "Condition" relation, there is a 35% increase in the actual

number of instances with 500 documents, and this figure jumps to over 150% with 2,000 documents. With these additional training instances, the gain in F-score for the "Condition" relation is 18.55% from Table 4.1. For the "Topic Change" relation, it is a pleasant surprise that there is a relative F-score improvement of 13.15% based on about 50% more training instances.

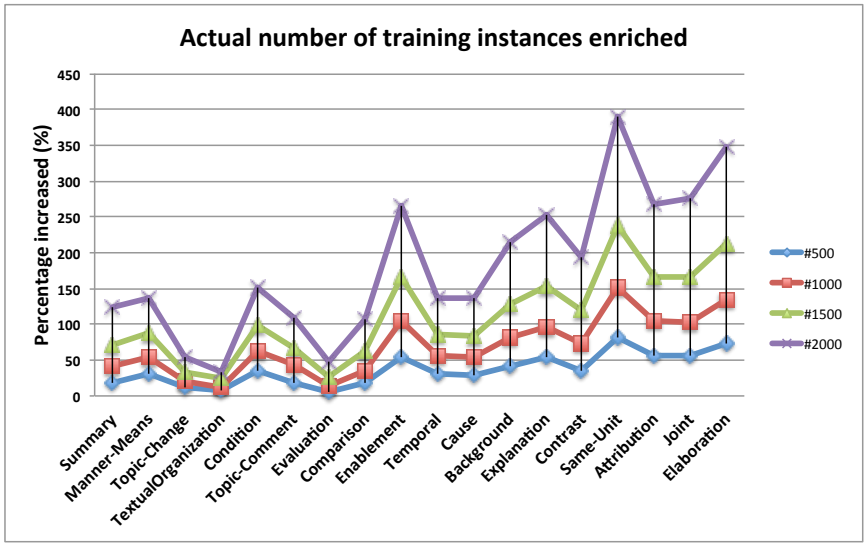


Figure 4.4: Actual Number of Training Instances Enriched (%)

The reader may wonder with 2000 more unlabeled documents, why there is only a modest increase in training instances for some of the infrequent relations. This increase of course depends on the filtering threshold. One temptation based on Table 4.1 is to lower the threshold to admit more training instances. This leads us to one of the most striking features of Figure 4.3 on how the relations are separated into two clusters. While there are improvements for the infrequent relations, there is no gain, or even small negative impact, on the frequent relations. This phenomenon clearly shows that co-training *without filtering* can be harmful to performance. The filtering step is essential to guard against adding "false positive" instances for re-training. If the filtering threshold is set too low, then the frequent relations may suffer. On the other hand, if the filtering threshold is set too high, then only few training instances will be added to benefit the infrequent relations.

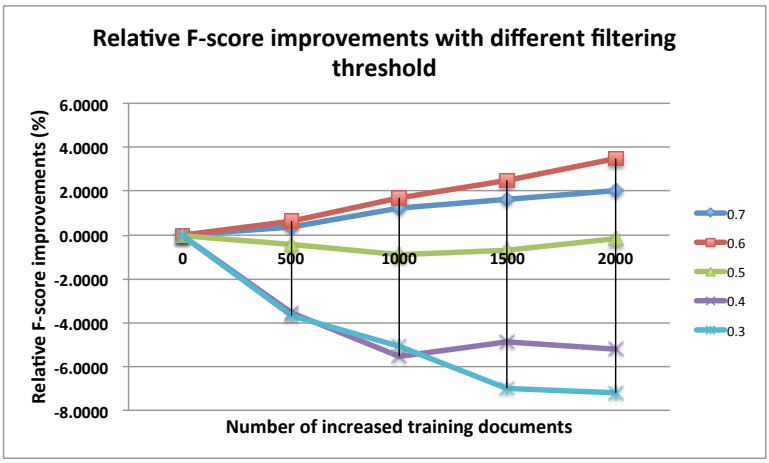
4.4 The Impact of the Filtering Threshold

The results presented so far are based on a filtering threshold of 0.5. To examine the impact of the filtering threshold on performance, we vary the threshold. Figure 4.5(a) shows how the relative F-score improvement changes with a filtering threshold from 0.3 to 0.7 aggregated across all the 18 relations. The results shown in the figure are based on all the instances in the entire dataset. In other words, the performance of the frequent relations, due to their much higher frequencies, completely dominates the performance of the infrequent ones. Thus, Figure 4.5(b) shows a corresponding graph aggregated across only the top-8 infrequent relations.

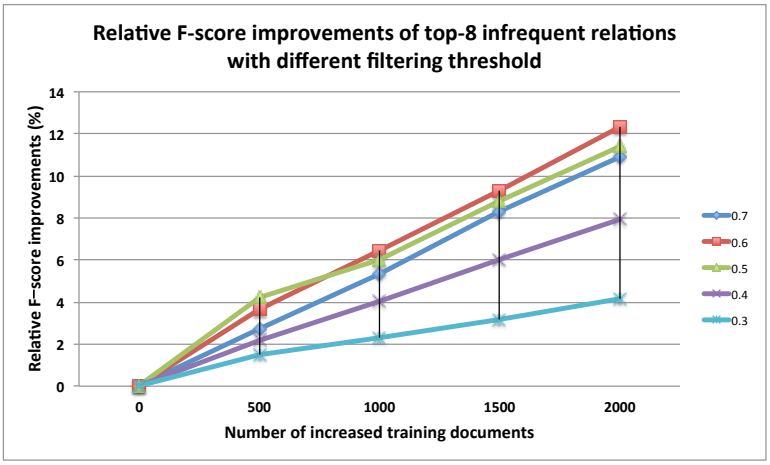
Compared with the filtering threshold of 0.5 shown previously, there is further improvement when the threshold is raised to 0.6 and 0.7. Particularly from Figure 4.5(b), there is considerable improvement across the top-8 infrequent relations. Interestingly, the peak performance gain occurs with the threshold of 0.6 – not 0.7. This shows that when the threshold is raised from 0.6 to 0.7, the reduction in the number of documents passing through the filter hurts the gain in performance.

The reader may wonder whether this kind of performance improvements will continue to grow under the effective threshold with more unlabeled resources added in. To explore the answer to this question, we employ the New York Times text corpus [26] by adding a small subset of its documents to our existing unlabeled documents. Then we conduct the same experiment with the expanded unlabeled resources, and the result in Figure 4.6 shows that the performance will continue to improve at a lower rate and finally tend to stabilize.

Next let us examine the situation when the filtering threshold is reduced from 0.5 to 0.4 and 0.3. Aggregated across all the 18 relations, Figure 4.5(a) clearly shows that there is performance loss. Consistent with the performance loss shown in Figure 4.3 for the frequent relations, this is the situation when the extra training instances passing through the filter introduce too much noise and hurt overall performance. Interestingly, Figure 4.5(b) shows that there is always a positive performance gain for the top-8 infrequent relations, regardless of whether the filtering threshold is 0.3 or 0.7. This suggests that infrequent relations and frequent relations may need different threshold. We will follow up on this heuristic in Section 4.7.



(a) Across All the 18 Relations



(b) Across the Top-8 Infrequent Relations

Figure 4.5: Changes in Relative F-score with Varying Filtering Agreement Score Threshold

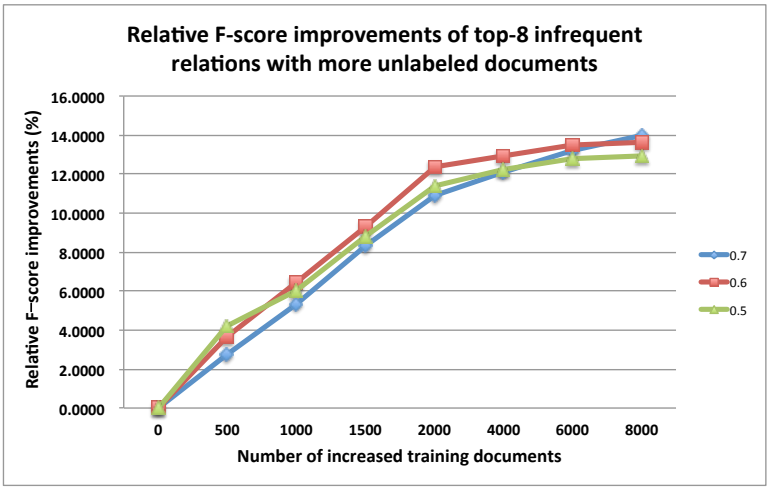


Figure 4.6: The Impact of More Unlabeled Resources

4.5 Using the Confidence Score to Approximate the Agreement Score

As discussed in Section 3, we explore a second form of enrichment. The agreement score reported so far requires the use of both the CODRA parser and the SR-parser. The former takes cubic time and the latter takes linear time. The idea here is to assess whether the confidence score generated from the faster SR-parser can be used to approximate the agreement score. If this approach is successful, an even larger number of unlabeled documents can be parsed rapidly to be used for re-training.

The first step of the assessment is to calculate the correlation between the agreement score and the confidence score of the SR-parser. As shown in Figure 4.7, which plots the correlation for all the 2,000 unlabeled documents, there is a weak correlation between the two scores. While the overall correlation is 0.36, it is promising to see that when the confidence score becomes higher (e.g., greater than 1.5), the correlation with the agreement score becomes stronger. It is also important to note that there is a significant drop in the number of documents passing the confidence score threshold of 2.

Corresponding to the two graphs in Figure 6, the two graphs in Figure 8 show the performance change using the confidence score of the SR-parser with varying

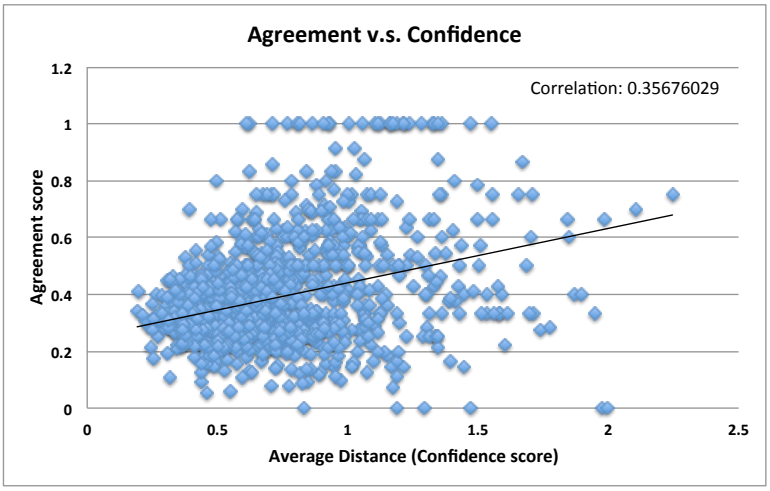
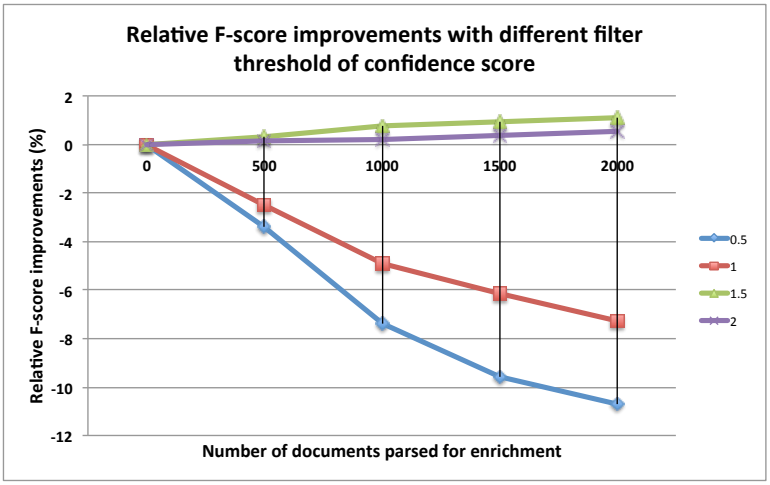


Figure 4.7: Agreement Score vs Confidence Score

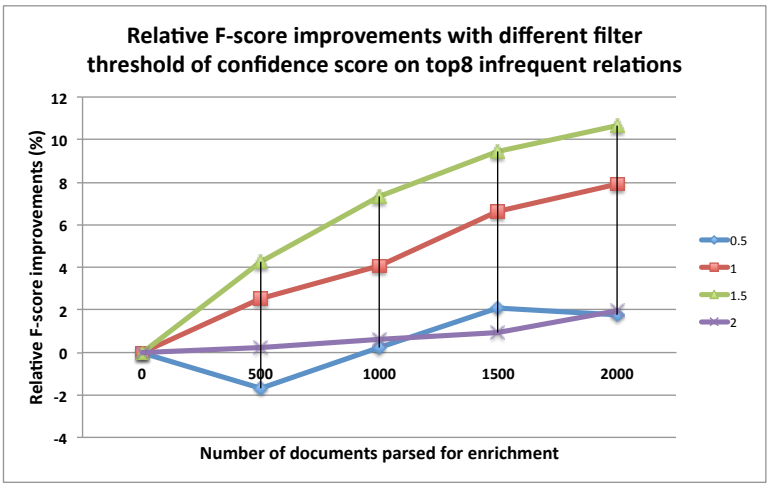
filtering threshold. Figure 4.8(a) shows how the relative F-score changes with a filtering threshold from 0.5 to 2 aggregated across all the 18 relations. Like in Figure 4.5(a) before, the performance of the frequent relations, due to their superior frequencies, completely dominates the performance of the infrequent ones. Thus, Figure 4.8(b) shows a corresponding graph aggregated across only the top-8 infrequent relations.

In Figure 4.8(b), the peak performance gain occurs when the confidence score threshold is 1.5. Even when the confidence score is lowered to 1.0, the performance gain is still reasonable with 2,000 documents. But somewhat surprisingly, the performance gain drops significantly when the confidence score threshold is raised to 2. This can be explained by looking more closely back at Figure 4.7. The confidence score threshold of 2 is too restrictive and very few unlabeled documents satisfy it; hence, the actual number of additional documents admitted for re-training is significantly reduced.

A first glance of Figure 4.8(a) seems to suggest that using the confidence score of the SR-parser is ineffective. The best performance gain across all the 18 relations is barely above 1%, which is smaller than the corresponding gain in Figure 4.5(a). This ineffectiveness is completely due to the behavior of the frequent relations. However, Figure 4.8(b) paints a rather different picture. For the top-8 infrequent



(a) On All 18 Relations



(b) On Top-8 Infrequent Relations

Figure 4.8: Overall F-score Improvements with Different Enriched Data Quality via Confidence Score

relations, there is a peak performance gain of about 10% with 2,000 documents. This gain is almost as good as the peak performance gain shown in Figure 4.5(b) with 2,000 documents. Given that the SR-parser is significantly faster than the CODRA parser, it is promising to use the confidence score of the SR-parser to approximate the agreement score, so that a larger number of unlabeled documents can be used for enrichment.

4.6 Adding Enriched Training Instances in an Iterative Manner

The results shown so far are based on one round of re-training. As shown in Figure 1, data enrichment can be done iteratively. The table below shows the relative F-score improvement on the top-8 infrequent relations when enrichment is done in increments of 500 documents. Here we process 500 unlabeled documents, re-train the SR-parser with the documents passing through the filter, then process the next batch of 500 documents, and so on.

# of documents	Basic	Iterative (batches of 500 documents)
1000	4.05	4.61
1500	6.65	7.47
2000	7.90	8.95

Table 4.2: Relative F-scores Improvements (%) on the Top-8 Infrequent Relations

The results shown in the table used the confidence score of 1 as the filtering threshold. The first column is precisely the curve in Figure 4.8(b) for the confidence score of 1. The first row in the table, for example, shows that doing re-training twice (500 documents each time) boosts the performance when compared with re-training done once at the end. Similarly, the other rows show that there is some value in iterative re-training.

4.7 Filtering at a Finer Granularity

All the filtering experiments above are done at the document level. That is, we calculate an agreement/confidence score for an entire discourse tree of a document, and if the score passes the threshold, this entire discourse tree along with every node on it are added to the new training instances, even though some nodes on this tree may have low scores. So in this section, we will explore the idea of filtering at a finer granularity, e.g. at the node level. Due to the different mechanism of the two parsers used in our framework, we picked the CODRA to conduct this experiment, because it is easier to filter discourse structures at node level and train its new model with partial discourse structures using CODRA. While we could not find a direct way to do it with the SR-parser.

In this experiment, we have performed both doc-level filtering and node-level filtering using the same experiment setting: we use the confidence score of CODRA itself to filter new candidate training examples, and the threshold is set to 0.5 here. The number of unlabeled documents used here is 500. The doc-level filtering works as described above, and for node-level filtering, every node with a confidence score higher than the threshold will be added to the new training set to retrain CODRA, no matter whether the document's discourse tree has a high confidence score that passes the threshold. Results of the two experiments are shown in Table 4.3. We can see that filtering at node-level has an advantage over filtering at doc-level for most discourse relations. And it is noteworthy that frequent relations are generally unharmed at node-level filtering, unlike at doc-level filtering.

Based on the control of filtering at a finer granularity, we can actually do more with the filtering threshold. Since in this case we can compare the score of each node to a threshold to determine whether it should be added to the new training set, we can actually set different thresholds for different types of relations. Though how to set different thresholds for different relations is still to be explored, we have run a small experiment with two different thresholds for infrequent and frequent relations separately and it shows a small increase on the performance. So we believe with more reasonable threshold set for different relations, in the future, greater improvements can be expected from using a varying threshold.

Relation	Doc-level	Node-level
Summary	4.265	6.811
Cause	1.827	1.965
Manner-Means	8.677	12.581
Temporal	-0.296	-0.246
Topic-Change	1.201	1.801
Background	-0.209	0.105
TextualOrganization	5.669	7.122
Explanation	-0.317	-0.106
Condition	6.656	7.488
Contrast	-0.066	0.131
Comparison	4.527	4.527
Same-Unit	-0.109	0.145
Evaluation	1.696	1.993
Attribution	-0.120	0.052
Topic-Comment	1.360	2.039
Joint	-0.211	-0.015
Enablement	2.999	3.314
Elaboration	-0.058	0.014

Table 4.3: Relative F-scores Improvements (%) at Different Filtering Granularities

Chapter 5

Conclusion

As the number of applications of discourse parsing in NLP is constantly growing, any improvement in discourse parsing performance can have considerable impact. In this thesis, we first validate the underfitting hypothesis, i.e., the less frequent a relation is in the training data, the poorer the performance on that relation. This is a phenomenon that applies to most discourse parser. One solution is, of course, to create more labeled data, ideally for all the relations. However, given the resources required for manually creating labeled data for discourse parsing, we explore in this thesis a training data enrichment framework that relies on co-training of the CODRA parser and the SR-parser on unlabeled documents. We also investigate using both the agreement score and the confidence score of the SR-parser to filter away "low quality" documents, whose presence in the re-training can hurt the performance. Our empirical results show that agreement score filtering can boost the performance of infrequent relations considerably. Our results also show that for infrequent relations, the confidence score of the SR-parser can also be used as a fast approximation of the agreement score.

So far our results show that our data enrichment framework is not effective for frequent relations. In ongoing work, we are studying how to augment our framework to boost the performance of even the frequent relations, and the varying threshold might be a promising solution. In the future, we plan to apply our framework to enrich training data for discourse structure and nuclearity analysis, and also to apply it to other discourse dataset(s) labeled in different ways (e.g. PDTB).

Bibliography

[1] S. Abney, S. Flickenger, C. Gdaniec, C. Grishman, P. Harrison, D. Hindle, R. Ingria, F. Jelinek, J. Klavans, M. Liberman, et al. Procedure for quantitatively comparing the syntactic coverage of english grammars. In *Proceedings of the workshop on Speech and Natural Language*, pages 306–311. Association for Computational Linguistics, 1991. → pages 21

[2] K. Allen, G. Carenini, and R. T. Ng. Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In *EMNLP*, pages 1169–1180, 2014. → pages 1, 3, 5

[3] P. Bhatia, Y. Ji, and J. Eisenstein. Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of the Empirical Methods in Natural Language Processing, (EMNLP)*, 2015. → pages 1, 3, 4

[4] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998. → pages 5, 13

[5] L. Carlson, D. Marcu, and M. E. Okurowski. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue - Volume 16*, SIGDIAL '01, pages 1–10, Stroudsburg, PA, USA, 2001. Association for Computational Linguistics. doi:10.3115/1118078.1118083. URL http://dx.doi.org/10.3115/1118078.1118083. → pages 2, 4

[6] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000. → pages 6

[7] V. W. Feng and G. Hirst. 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, 2012. → pages 10

[8] V. W. Feng and G. Hirst. A linear-time bottom-up discourse parser with constraints and post-editing. In *ACL (1)*, pages 511–521, 2014. → pages 4, 11, 16

[9] S. Gerani, Y. Mehdad, G. Carenini, R. T. Ng, and B. Nejat. Abstractive summarization of product reviews using discourse structure. In *EMNLP*, pages 1602–1613, 2014. → pages 3

[10] S. Gerani, G. Carenini, and R. T. Ng. Modeling content and structure for abstractive review summarization. *Computer Speech & Language*, 2016. → pages 1, 3

[11] F. Guzmán, S. Joty, L. Màrquez, and P. Nakov. Using discourse structure improves machine translation evaluation. In *ACL (1)*, pages 687–698, 2014. → pages 1, 3

[12] H. Hernault, D. Bollegala, and M. Ishizuka. A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 399–409. Association for Computational Linguistics, 2010. → pages 4, 13

[13] H. Hernault, H. Prendinger, D. A. DuVerle, M. Ishizuka, and T. Paek. Hilda: a discourse parser using support vector machine classification. *Dialogue and Discourse*, 1(3):1–33, 2010. → pages 10

[14] L. Horn. *A natural history of negation*. Chicago: University of Chicago Press, 1989. → pages 5

[15] Y. Ji and J. Eisenstein. Representation learning for text-level discourse parsing. In *ACL (1)*, pages 13–24, 2014. → pages 4, 6, 10, 11, 15

[16] S. Joty, G. Carenini, and R. T. Ng. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*, 2015. → pages 1, 2, 4, 6, 10, 12, 15

[17] S. R. Joty, G. Carenini, R. T. Ng, and Y. Mehdad. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *ACL (1)*, pages 486–496, 2013. → pages 10, 11

[18] M. Lan, Y. Xu, Z.-Y. Niu, et al. Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In *ACL (1)*, pages 476–485. Citeseer, 2013. → pages 12

[19] J. J. Li and A. Nenkova. Fast and accurate prediction of sentence specificity. In *AAAI*, pages 2281–2287, 2015. → pages 5, 14

[20] Y. Liu, S. Li, X. Zhang, and Z. Sui. Implicit discourse relation classification via multi-task neural networks. *arXiv preprint arXiv:1603.02776*, 2016. → pages 12, 13

[21] D. Marcu. A decision-based approach to rhetorical parsing. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 365–372. Association for Computational Linguistics, 1999. → pages 9

[22] D. Marcu. *The theory and practice of discourse parsing and summarization*. MIT press, 2000. → pages 6, 17, 21

[23] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993. → pages 21

[24] K. McKeown and O. Biran. 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, 2013. → pages 12

[25] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. K. Joshi, and B. L. Webber. The penn discourse treebank 2.0. In *LREC*. Citeseer, 2008. → pages 4

[26] E. Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008. → pages 26

[27] R. Soricut and D. Marcu. Sentence level discourse parsing using syntactic and lexical information. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 149–156. Association for Computational Linguistics, 2003. → pages 9

[28] C. Sporleder and A. Lascarides. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(3):369–416, 2008. → pages 12

[29] R. Subba and B. Di Eugenio. An effective discourse parser that uses rich linguistic information. In *Proceedings of Human Language Technologies:*

The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 566–574. Association for Computational Linguistics, 2009. → pages 10

[30] X. Wan. Co-training for cross-lingual sentiment classification. 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 1-Volume 1*, pages 235–243. Association for Computational Linguistics, 2009. → pages 13

[31] Z.-M. Zhou, Y. Xu, Z.-Y. Niu, M. Lan, J. Su, and C. L. Tan. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1507–1514. Association for Computational Linguistics, 2010. → pages 12