Training Data Enrichment for Infrequent Discourse Relations

Kailang Jiang, Giuseppe Carenini, Raymond T. Ng
Department of Computer Science
University of British Columbia
Vancouver, BC, Canada, V6T 1Z4
{jiangkl, carenini, rng}@cs.ubc.ca

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 paper, 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.

1 Introduction

Discourse parsing is widely used in text understanding (Allen et al., 2014), sentiment analysis (Bhatia et al., 2015) and other NLP tasks (Guzmán et al., 2014) (Gerani et al., 2016). 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 benchmark RST-DT dataset (Carlson et al., 2001) uses 18 discourse relations. Studies in the past decade on discourse parsing, such as (Ji and Eisenstein, 2014), (Feng and Hirst, 2014), and (Joty et al., 2015), greatly improved the performance of discourse parsing in general. However, it has been observed that the performance across the discourse relations varies significantly (Joty et al., 2015), and that poor performance may be linked to underfitting, i.e., a lack of training data (Feng and Hirst, 2014).

In this paper, we investigate the underfitting hypothesis and study how to improve the situation. 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 very popular corpus — Rhetorical Structure Theory Discourse Treebank (RST-DT) (Carlson et al., 2001) 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 (Prasad et al., 2008), 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 (Hernault et al., 2010). Given that the performance 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

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2603–2614, Osaka, Japan, December 11-17 2016.

ones. In Section 4, 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 (Horn, 1989) (Allen et al., 2014). Moreover, the "Instantiation" relation from PDTB is regarded as an important feature for sentence specificity prediction (Li and Nenkova, 2015).

The main objective of this paper 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. 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 (Li and Nenkova, 2015) and information retrieval (Blum and Mitchell, 1998).

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 underfitting 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 is shown in Figure 1, when it is applied to two discourse parsers, P1 and P2. P1 and P2 are initially trained on labeled data and then are applied to unlabeled data to generate new high-quality training examples for further re-training.

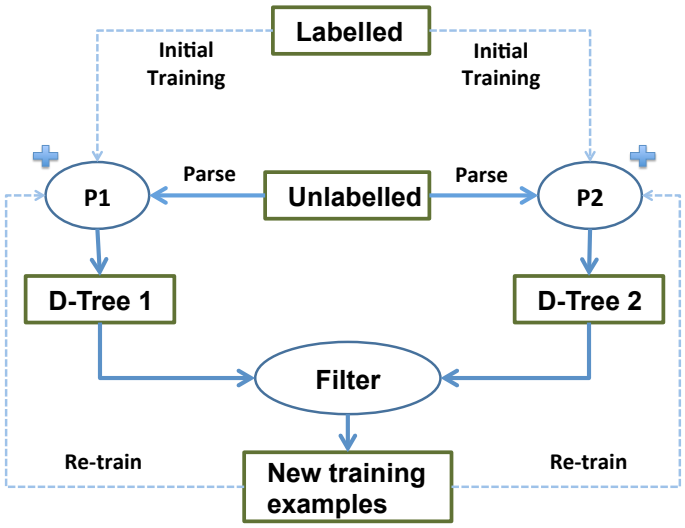


Figure 1: Workflow of our enrichment approach

The specific contributions of our paper 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 (Dietterich, 2000), we choose two very different discourse parsers, namely the CKY-like CODRA parser by (Joty et al., 2015) and the Shift-Reduce (SR) parser by (Ji and Eisenstein, 2014). While Section 3 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 (Marcu, 2000). Only the discourse relation instances in discourse trees with high-enough agreement scores pass through the filter for re-training purposes. Section 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, Section 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.

2 Related work

Recent discourse parsers have improved the overall performance of discourse parsing in different ways. (Joty et al., 2013) (Joty et al., 2015) proposed a two-stage document-level discourse parser CODRA, which builds a discourse tree by applying an optimal parsing algorithm to probabilities inferred from two Conditional Random Fields: one for intrasentential parsing and the other for multisentential parsing. This approach achieves good performance in discourse relation labeling. Based on their idea, (Feng and Hirst, 2014) developed a similar but much faster model that adopts a greedy bottom-up approach, with two linear-chain CRFs applied in cascade as local classifiers. On the other hand, (Ji and Eisenstein, 2014) proposed a Shift-Reduce (SR) parser that combines the machinery of large-margin transition-based structured prediction with representation learning. This method also reports a good overall performance with linear running time. However, all these state-of-the-art discourse parsers still perform badly on infrequent relations due to insufficient training examples.

The problem of lacking training examples also impacts other aspects of discourse parsing, for example parsing 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) 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 and Biran, 2013) tackles the feature sparsity problem by aggregating implicit relations into larger groups. And recently (Lan et al., 2013) combines different data through multi-task learning. The method performs implicit and explicit relation classification in the PDTB framework as two tasks and relies on multi-task learning to obtain higher performance.

(Liu et al., 2016) 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 relatively infrequent relations, though they only apply their scheme on the four coarse top-level relations. 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 (Sporleder and Lascarides, 2008). Moreover, (Joty et al., 2015) has shown that there are many more features beyond cue phrases that are useful for discourse parsing. (Hernault et al., 2010) proposes a feature vector extension approach to improve classification of infrequent discourse relations. The approach is based on word co-occurrence. Partly because a simple discourse parser was used, their approach is shown to produce only minimal improvements in performance.

Unlike (Liu et al., 2016) and (Hernault et al., 2010), 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. We employ the idea of *co-training*, which is first introduced by (Blum and Mitchell, 1998) with its application in helping the search engine better classify "academic course home page". Similar co-training efforts have been found effective in many NLP problems when only a small amount of labeled data is available. For example, (Wan, 2009) proposes a co-training approach for cross-lingual

sentiment classification, while (Li and Nenkova, 2015) applies co-training on predicting sentence specificity.

3 Our Enrichment Approach

The workflow of our enrichment approach is shown in Figure 1. 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.

In our approach, the first parser we pick is the CODRA parser (Joty et al., 2015), 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 (Ji and Eisenstein, 2014), 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 discourse parser (Feng and Hirst, 2014) 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.

Co-training alone, however, is not sufficient. Since both the CODRA parser and the SR-parser performs poorly on 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 paper we investigate two forms of enrichment, based on the agreement score between the two parsers, and the confidence score given by the SR-parser.

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, because in general the CODRA parser is more accurate than the SR-parser. After that, we treat the parse tree produced by the SR-parser as testing, and use the F-score for comparing discourse trees proposed in (Marcu, 2000) 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 paper is based on using the confidence score of the SR-parser as an approximation of the agreement score between the two parsers. The SR-parser 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.

Furthermore, we can also control the filtering threshold score at a finer granularity. That is, if the parser allows a user to feed a partial discourse tree or even just some nodes for training, we can set the filtering threshold on each node instead of each document. In this case, if a discourse tree has a high agreement/confidence score, but some nodes on the tree only have a low score, instead of adding the entire discourse tree to the new training set, we can remove those nodes with low score and add only those nodes with high scores to the new training set. This way, we can filter at a finer granularity. Moreover, based on this node-level filtering framework, we can actually set different threshold for different types of nodes, for example, we can set a lower threshold for infrequent relations, and a higher one for frequent relations.

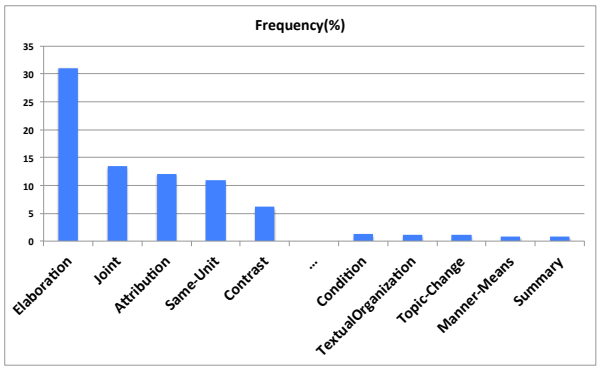


Figure 2: Distribution of the most frequent and the least frequent 5 relations in RST-DT

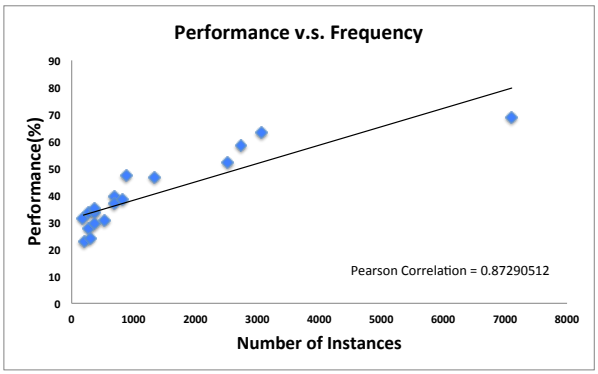


Figure 3: Performance versus Frequency for each relation

4 Empirical Evaluation

4.1 Datasets

In this paper, we use the RST-DT dataset as the gold standard labeled data. It consists of 385 documents selected from Penn Treebank (Marcus et al., 1993), which are all originally articles from the Wall Street Journal. Those 385 documents were divided by the author into two groups: the training set consisting of 347 documents, and the test set 38 documents. For results reported in this paper, 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 re-trained using the enriched dataset.

For the unlabeled documents, we used 2000 Wall Street Journal articles from the Penn Treebank dataset (Marcus et al., 1993). 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 (Abney et al., 1991). More specifically, we use the F-score measure for comparing discourse trees, as proposed in (Marcu, 2000).

4.2 The Underfitting Hypothesis: Performance vs Frequency

As for the discourse relations, we examine all the 18 coarse-grained relations defined in (Carlson et al., 2001). Figure 2 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%.

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 3 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.

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 1 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 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.

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 1: Relative F-scores improvements (%) on the top-8 infrequent relations

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.

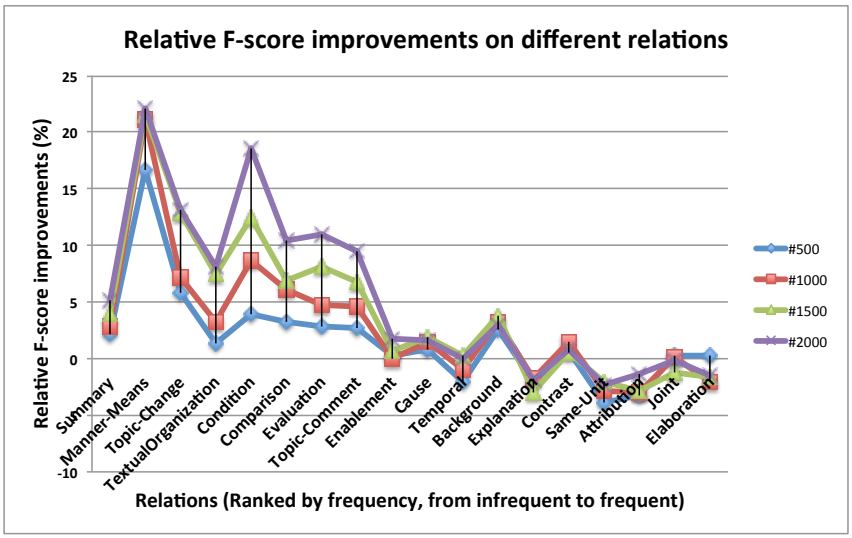


Figure 4: Relative F-score improvements on different relations

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 5 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 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.

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 1 is to lower the threshold to admit more training instances. This leads us to one of the most striking features of Figure 4 on how the relations are separated into

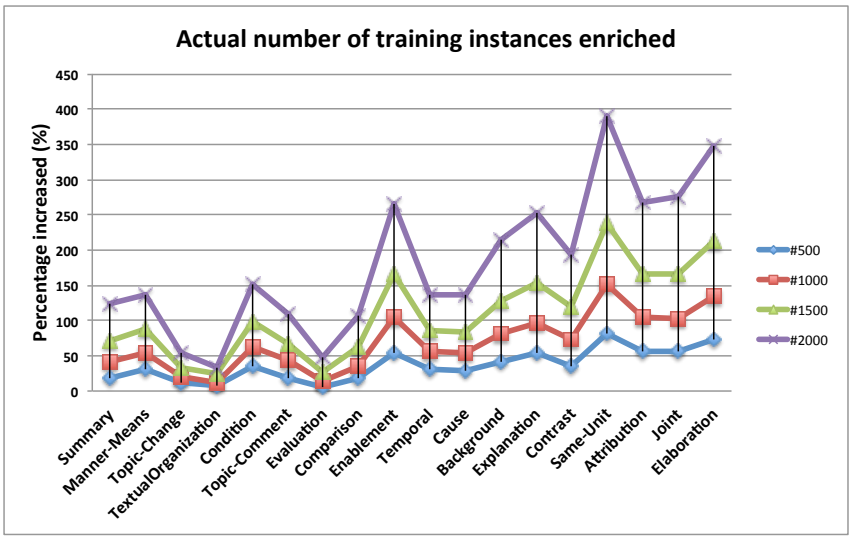
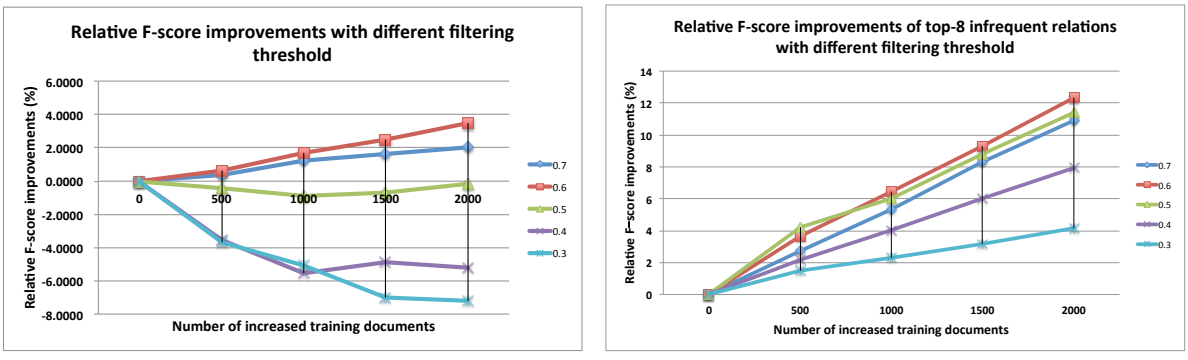


Figure 5: Actual number of training instances enriched (%)

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 6(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 6(b) shows a corresponding graph aggregated across only the top-8 infrequent relations.



(a) Across all the 18 relations

(b) Across the top-8 infrequent relations

Figure 6: Changes in relative F-score with varying filtering agreement score threshold

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 6(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 (Sandhaus, 2008) 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 7 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 6(a) clearly shows that there is performance loss. Consistent with the performance loss shown in Figure 4 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 6(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.

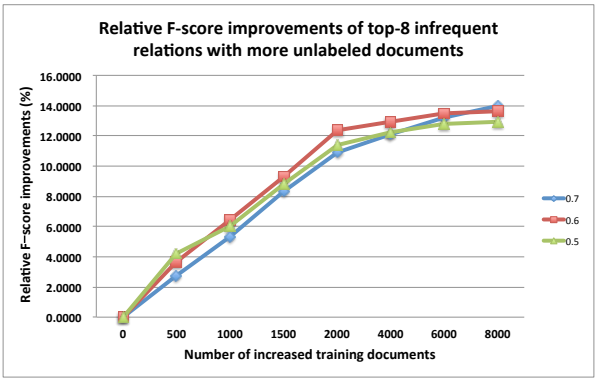


Figure 7: The impact of more unlabeled resources

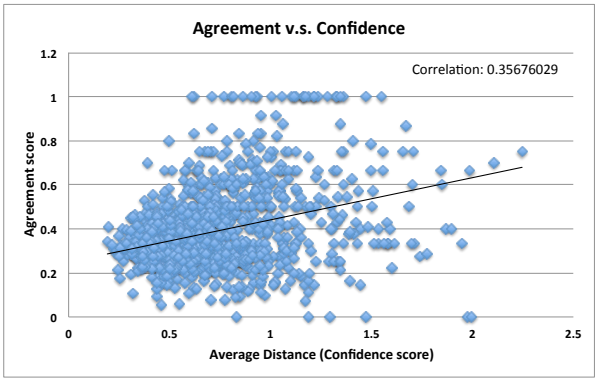


Figure 8: Agreement score v.s. confidence score

4.5 Using the Confidence Score to Approximate the Agreement Score

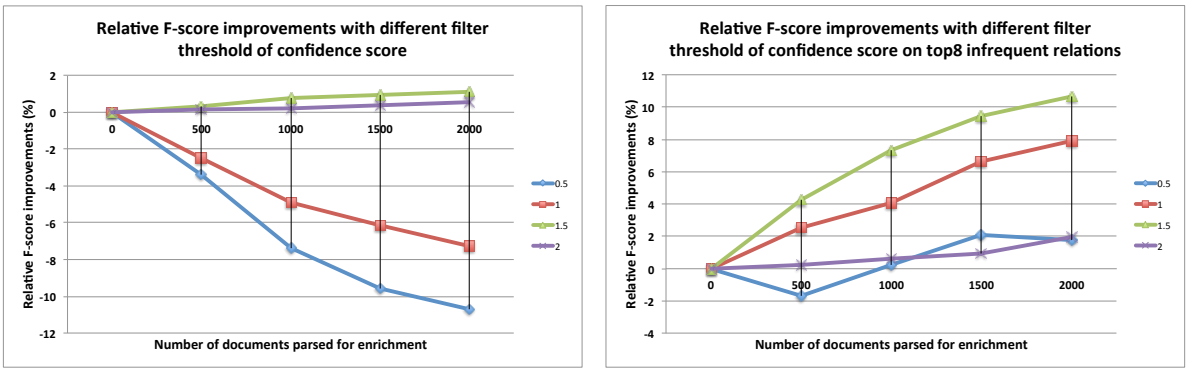
As discussed in Section 3, we explore a second form of enrichment. The agreement score reported so far requires the use of both the CODRA parser and the SR-parser. The former takes cubic time and the latter takes linear time. The idea here is to assess whether the confidence score generated from the 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 8, 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 9 show the performance change using the confidence score of the SR-parser with varying filtering threshold. Figure 9(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 6(a) before, the performance of the frequent relations, due to their superior frequencies, completely dominates the performance of the infrequent ones. Thus, Figure 9(b) shows a corresponding graph aggregated across only the top-8 infrequent relations.

In Figure 9(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 8. 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) On all 18 relations

(b) On top-8 infrequent relations

Figure 9: Overall F-score improvements with different enriched data quality via confidence score

A first glance of Figure 9(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 6(a). This ineffectiveness is completely due to the behavior of the frequent relations. However, Figure 9(b) paints a rather different picture. For the top-8 infrequent 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 6(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 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 9(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 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. 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 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.

Relation	**Doc-level**	**Node-level**	**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 3: Relative F-scores improvements (%) at different filtering granularities

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.

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 paper, 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 paper 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 enrichment 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).

References

Steven Abney, S Flickenger, Claudia Gdaniec, C Grishman, Philip Harrison, Donald Hindle, Robert Ingria, Frederick Jelinek, Judith Klavans, Mark Liberman, et al. 1991. 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.

Kelsey Allen, Giuseppe Carenini, and Raymond T Ng. 2014. Detecting disagreement in conversations using pseudo-monologic rhetorical structure. In *EMNLP*, pages 1169–1180.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of the Empirical Methods in Natural Language Processing, (EMNLP)*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurowski. 2001. 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. Association for Computational Linguistics.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *ACL (1)*, pages 511–521.

Shima Gerani, Giuseppe Carenini, and Raymond T Ng. 2016. Modeling content and structure for abstractive review summarization. *Computer Speech & Language*.

Francisco Guzmán, Shafiq Joty, Lluís Màrquez, and Preslav Nakov. 2014. Using discourse structure improves machine translation evaluation. In *ACL (1)*, pages 687–698.

Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. 2010. 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.

Laurence Horn. 1989. *A natural history of negation*. Chicago: University of Chicago Press.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *ACL (1)*, pages 13–24.

Shafiq R Joty, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining intra-and multi-sentential rhetorical parsing for document-level discourse analysis. In *ACL (1)*, pages 486–496.

Shafiq Joty, Giuseppe Carenini, and Raymond T Ng. 2015. Codra: A novel discriminative framework for rhetorical analysis. *Computational Linguistics*.

Man Lan, Yu Xu, Zheng-Yu Niu, et al. 2013. Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In *ACL (1)*, pages 476–485. Citeseer.

Junyi Jessy Li and Ani Nenkova. 2015. Fast and accurate prediction of sentence specificity. In *AAAI*, pages 2281–2287.

Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. 2016. Implicit discourse relation classification via multi-task neural networks. *arXiv preprint arXiv:1603.02776*.

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. MIT press.

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.

Kathleen McKeown and Or Biran. 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.

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.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Caroline Sporleder and Alex Lascarides. 2008. Using automatically labelled examples to classify rhetorical relations: An assessment. *Natural Language Engineering*, 14(3):369–416.

Xiaojun Wan. 2009. 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.

Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. 2010. 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.