What Causes the Failure of Explicit to Implicit Discourse Relation Recognition?

Wei Liu[1], **Stephen Wan**[2], **Michael Strube**[1]
[1]Heidelberg Institute for Theoretical Studies gGmbH
[2]CSIRO Data61
wei.liu@h-its.org | stephen.wan@csiro.au | michael.strube@h-its.org

Abstract

We consider an unanswered question in the discourse processing community: why do relation classifiers trained on explicit examples (with connectives removed) perform poorly in real implicit scenarios? Prior work claimed this is due to linguistic dissimilarity between explicit and implicit examples but provided no empirical evidence. In this study, we show that one cause for such failure is a label shift after connectives are eliminated. Specifically, we find that the discourse relations expressed by some explicit instances will change when connectives disappear. Unlike previous work manually analyzing a few examples, we present empirical evidence at the corpus level to prove the existence of such shift. Then, we analyze why label shift occurs by considering factors such as the syntactic role played by connectives, ambiguity of connectives, and more. Finally, we investigate two strategies to mitigate the label shift: filtering out noisy data and joint learning with connectives. Experiments on PDTB 2.0, PDTB 3.0, and the GUM dataset demonstrate that classifiers trained with our strategies outperform strong baselines.

1 Introduction

Discourse relations, such as *Contrast* and *Cause*, describe the logical relationship between two text spans (i.e., arguments). They can either be signaled explicitly with connectives, as in (1), or expressed implicitly, as in (2):

(1) [They may feel emotionally secure now]$_{\text{Arg1}}$ **<u>because</u>** [they are not heavily in the stock market.]$_{\text{Arg2}}$
—— *Contingency.Cause*

(2) [He has not changed, but those around him have.]$_{\text{Arg1}}$ [Many of his view on the protection of wilderness areas are now embraced by mainstream.]$_{\text{Arg2}}$
—— *Contingency.Cause*

Corpora of explicit discourse relations are relatively easy to create manually and automatically (Marcu and Echihabi, 2002) since connectives are strong cues for identifying relations (Pitler and Nenkova, 2009). In contrast, the annotation of implicit relations is hard and expensive because one must infer the sense from input texts. This has prompted many early studies (Marcu and Echihabi, 2002; Lapata and Lascarides, 2004; Sporleder and Lascarides, 2005; Saito et al., 2006) to use explicit examples to classify implicit relations (dubbed **explicit to implicit relation recognition**). The main idea is to construct an *implicit-like* corpus by removing connectives from explicit instances, and use it to train a classifier for implicit relation recognition. While this method attains good results on test sets constructed in the same manner, it is reported by Sporleder and Lascarides (2008) to perform poorly in real implicit scenarios. They claim this is caused by the linguistic dissimilarities between explicit and implicit examples, but provide no corpus-level empirical evidence. More recent works (Huang and Li, 2019; Kurfalı and Östling, 2021) focus on enhancing the transfer performance from explicit to implicit discourse relations. However, little attention has been paid to the underlying causes of the poor results.

In this paper, we show that one cause for the poor transfer performance is the occurrence of label shift during the construction of the *implicit-like* corpus. Removing connectives from explicit examples affects the discourse relations they originally expressed. Referring to example (3), which contains the connective *then* and is annotated as a *Temporal.Asynchronous* relation:

(3) [Crossland Savings Bank's stock plummeted.]$_{\text{Arg1}}$ **<u>Then</u>** [management recommended a suspension of dividend payments on both its common and preferred stock.]$_{\text{Arg2}}$
—— *Temporal.Asynchronous*

When the connective *then* is removed, the example, however, tends to express a *Contingency.Cause* relation because the first argument describes a re-

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 2738–2753

sult of "stock plummet" and the second argument gives the reason, a suspension of dividend pay. To verify the existence of the label shift, we first manually analyze a small number of explicit examples with connectives removed, and summarize different cases of instances suffering such shift. Then, we provide empirical evidence to demonstrate that label shift is present not only in a few examples but at the corpus level. Consequently, classifiers trained on the *implicit-like* corpus learn a chaotic pattern for relation classification (i.e., being taught to predict an example with *Cause* relation as *Asynchronous* relation as in the above example), resulting in poor performance in real implicit scenarios. We further analyze why label shift happens in the *implicit-like* corpus by considering factors such as the syntactic role played by connectives, ambiguity of connectives, and more. Our results reveal that the syntactic role played by connectives contributes the most to the occurrence of the label shift.

Based on this observation, we investigate two strategies from both the data and training aspects to alleviate the influence of the label shift. We devise a label shift metric to quantify the degree of label shift that occurs in each explicit example and employ it for sample-level filtering. Additionally, we study a joint-learning strategy from the training side to further alleviate the impact of the shift in cases where the filtering results are imperfect. Specifically, our classifier jointly learns to recover a connective from arguments and identify a relation based on the recovered connective and arguments.

We evaluate the effectiveness of our approach on two datasets: the Penn Discourse Treebank 2.0 (PDTB 2.0, Prasad et al., 2008) and 3.0 (PDTB 3.0, Webber et al., 2019b). Experiments show that our model improves the performance of explicit to implicit discourse relation recognition, achieving encouraging results on both datasets. Furthermore, to test the generalizability of the proposed method, we conduct experiments on the GUM dataset (Zeldes, 2017), which is annotated with relations from Rhetorical Structure Theory (RST, Mann and Thompson, 1988). The results suggest that our filtering mechanism and joint training strategy also help with the explicit to implicit relation classification on the GUM dataset.

2 Related Work

Learning to use lexically-marked examples to classify implicit relations has received continued research attention. Marcu and Echihabi (2002) train the first classifier for implicit intra-sentential discourse relations using explicitly-marked examples from a raw English corpus, BLIPP (Charniak, 2000), and the RST Treebank (Carlson et al., 2001). Lapata and Lascarides (2004) present a similar approach using BLIPP but focus on sentence-internal temporal relations. Blair-Goldensohn et al. (2007) extend this work by refining the training process using parameter optimization, topic segmentation, and syntactic parsing on the Gigaword (Graff et al., 2003) and PDTB (Prasad et al., 2006b). These three works are evaluated on test sets constructed in the same manner as the training set and show good performance. Sporleder and Lascarides (2008) and Lin et al. (2009) investigate the applicability of this approach to real implicit scenarios and find its performance is poor. They claim, based on manual analysis of a few instances, that the linguistic dissimilarities between explicit and implicit examples may be the cause. However, a corpus-level empirical analysis is not provided to establish how widespread the problem is.

More recent work has focused on improving the performance from explicit to implicit discourse relation recognition. Wang et al. (2012) propose to use typical examples with linguistic structure shared between explicit and implicit relations for training. Ji et al. (2015) adopt techniques such as resampling from transfer learning to handle the mismatched label distribution between explicit and implicit corpora. Huang and Li (2019) follow a similar domain adaptation idea but focus on minimizing the distance between representations of explicit and implicit examples with an adversarial training framework. Kurfalı and Östling (2021) tackle this task from a distant-supervision perspective. In contrast to the above work, we aim to present a new understanding of the question "Why classifiers trained on explicit examples perform poorly in real implicit scenarios" and to provide corpus-level empirical evidence to support our findings.

Connective information has been widely studied in discourse relation recognition. Pitler and Nenkova (2009) train a classifier with connectives in the text as the only features and find it could achieve over 90% accuracy on explicit relation recognition. Similarly, many attempts have been made using connectives to improve the recognition performance on implicit relations, including pipeline methods (Zhou et al., 2010; Jiang et al.,

ID	Label Shift	Text	Relation
(4)	Yes	[We backed this bill because we thought it would help Skinner]$_{Arg1}$ [now we're out there dangling in the wind.]$_{Arg2}$	Comparison.Contrast
		[We backed this bill because we thought it would help Skinner]$_{Arg1}$ **and** [now we're out there dangling in the wind.]$_{Arg2}$	Expansion.Conjunction
(5)	No	[The procedure causes great uncertainty]$_{Arg1}$ [an investor can't be sure of his or her individual liability.]$_{Arg2}$	Contingency.Cause
		[The procedure causes great uncertainty]$_{Arg1}$ **because** [an investor can't be sure of his or her individual liability.]$_{Arg2}$	Contingency.Cause

Figure 1: Examples of suffering and not suffering the label shift.

2021), multi-task training (Kishimoto et al., 2020; Long and Webber, 2022), adversarial training (Qin et al., 2017), joint training (Liu and Strube, 2023), and prompt learning (Zhou et al., 2022; Xiang et al., 2023). Our work differs from them in both motivation and application scenarios. For example, we use connective information not as a feature to the classifier but as a filtering mechanism to select good training instances.

3 Experimental Setup

We introduce the task of explicit to implicit relation recognition and describe the experimental setup[1] used for analyses in Section 4 and improvements in Section 6.

Task. The task of explicit to implicit relation recognition builds an implicit classifier relying on explicit examples. The traditional approach to achieving this is to construct an *implicit-like* corpus by excluding connectives from explicit examples, and then train a classifier on this corpus with the original explicit relations as ground-truth labels.

Datasets. The datasets we use for analyses are PDTB 2.0 (Prasad et al., 2008) and PDTB 3.0 (Webber et al., 2019b). PDTBs are corpora annotated with a lexical-based framework where instances are divided into different groups, including the discourse relation categories we focus on: explicit and implicit. This clear grouping makes them very suitable for explicit to implicit relation recognition (Huang and Li, 2019; Kurfalı and Östling, 2021) since we do not need to distinguish explicit or implicit examples by ourselves. More importantly, the two corpora offer manually annotated connectives for implicit examples (See Appendix A.1), facilitating our comparative analysis of explicit and implicit relations.

We follow previous work (Huang and Li, 2019) to use PDTB sections 2-20, 0-1, and 21-22 as training, development, and test set, respectively. We conduct experiments on both the top- and second-level relations of the two corpora.

Models. The relation classifier employed in this paper, including models for analysis in section 4 and baselines in section 6, consists of a pre-trained encoder and a linear layer. We follow previous work (Zhou et al., 2022; Long and Webber, 2022) to use $\text{RoBERTa}_{\text{base}}$ as the encoder. We show in Appendix B.2 that our findings are consistent across different pre-trained models and sizes. See Appendix A for more detailed settings.

4 Label Shift in Discourse Relations

4.1 What is label shift?

We consider label shift as the difference in relations observed between the same example with and without a connective:

$$\text{Rel}(\text{Arg1}, \text{Conn}, \text{Arg2}) \neq \text{Rel}(\text{Arg1}, \text{Arg2}) \quad (1)$$

where Arg1 and Arg2 are arguments of the example, and Conn denotes the connective. Figure 1 shows examples of suffering and not suffering the label shift. Example (4) was originally annotated as an *Expansion.Conjunction* relation because of the connective *and*. When *and* is removed, the example tends to express a *Comparison.Contrast* relation because of the contrast in lexical cues (e.g., "would help" vs. "dangling in the wind"). Regarding example (5), the arguments express the same relation as the connective *because* since the first argument describes a result of "uncertain" and the second argument presents the reason, i.e., unsure of liability.

4.2 Do explicit examples suffer label shift?

We manually analyze 100 explicit instances in PDTB 2.0 to ascertain the existence of label shift. Specifically, we randomly sample 100 explicit examples from the PDTB 2.0 and remove the connectives. Then, we train two students to annotate raw texts with PDTB relations. After finishing the training, the two students are asked to annotate those 100 examples (with connective removed), and the

[1] https://github.com/liuwei1206/Exp2Imp

(6) [Mr. Stein and other officers decided to sell that business]$_{Arg1}$ **after** [Japanese competitors grabbed a dominant share of the market.]$_{Arg2}$

—— *Temporal.Asynchronous*

(7) [There's nothing in the least contradictory in all this]$_{Arg1}$ **and** [it would be nice to think that Washington could tolerate a reasonably sophisticated, complex view.]$_{Arg2}$

—— *Expansion.Conjunction*

Figure 2: Different cases suffering label shift.

inter-annotator agreement is 0.7346 calculated in Cohen's Kappa.[2] See Appendix B.1 for more annotation information.

We find that 37 of these 100 examples were annotated with relations different from the original annotation, suggesting the existence of the label shift. We identify three different cases of suffering label shift: (i) Removing connectives leads to different relations. Referring to example (3), where the connective *then* signals a *Temporal* relation while the arguments express a *Contingency* relation because the first argument describes a result of "stock plummet" and the second one points out the reason, a suspension of dividend pay. (ii) Deleting connectives causes ambiguity in relations. This occurs when arguments contain clues to multiple relations without favoring a certain one. Considering example (6) in Figure 2, the arguments can express *Contingency* or *Temporal* relations since inserting *because* or *after* between arguments is acceptable. (iii) No relation is observed after eliminating the connective. This happens when there are no clues indicating discourse relations or arguments are too short to provide sufficient context. Referring to examples (7) in Figure 2, there is low lexical cohesion between the two arguments, requiring extensive world knowledge to understand that "Washington" refers to the U.S. government and "politics" can be "complex" or "contradictory", making it hard to infer any relation.

4.3 Does this shift exist at the corpus level?

We devise an empirical approach to show that label shift exists at the corpus level. The key idea comes from our definition of label shift, where an example is considered as suffering shift if its expressed relations are different when containing or not containing a connective. We mimic this judgment process but replace relations inferred by humans with those predicted by relation classifiers. Specifically, given

[2]We use cohen_kappa_score in sklearn.

Algorithm 1 Measuring Label Shift

Input: Relation Classifier $\mathbf{M}$, Corpus with Connectives $\{(\text{Arg1}_i, \text{Conn}_i, \text{Arg2}_i, \text{Rel}_i)\}|_{i=1}^{N}$
Output: diff_num, scores

```
Train(M, {(Arg1_i, Arg2_i, Rel_i)}|_{i=1}^{N})
diff_num = 0
scores = []
for i = 1, ..., N do
    # without and with connectives
    p1 = M.pred(Arg1_i, Arg2_i)
    p2 = M.pred(Arg1_i, Conn_i, Arg2_i)
    v1 = M.get_rep(Arg1_i, Arg2_i)
    v2 = M.get_rep(Arg1_i, Conn_i, Arg2_i)
    if p1 ≠ p2 then
        diff_num = diff_num + 1
    end if
    value = cosine_similarity(v1, v2)
    Append(scores, value)
end for
```

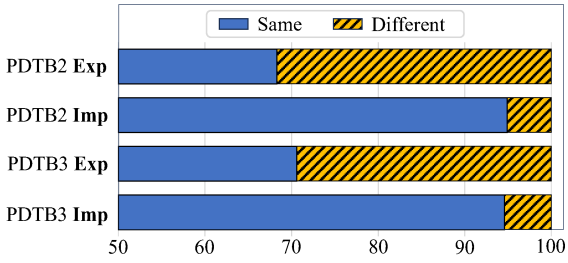
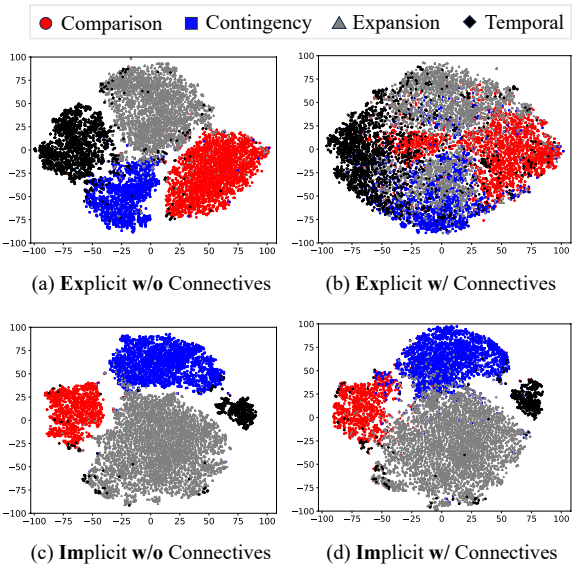


Figure 3: Percentage of examples in **Exp**licit and **Imp**licit corpora that receive the same and different predictions when containing and not containing a connective.

a corpus with connectives (either explicit corpus or implicit corpus with implicit connectives), we first train a relation classifier using arguments-label pairs[3] of the corpus. Then, we compare the classifiers' predictions on this corpus with and without the use of connectives (i.e., explicit examples vs. explicit examples with connectives removed, or implicit examples with connectives vs. implicit examples). If the predictions in the two settings are very different (see diff_num in Algorithm 1), it implies that connectives can substantially affect the semantics of examples throughout the corpus. That is, label shift exists across the entire dataset.

We conduct analyses on both explicit and implicit parts of PDTB 2.0 and 3.0, providing a comparison between these two types of examples. Figure 3 shows the assessment results on PDTB 2.0 and 3.0 (on top-level relations). In explicit corpora, connectives are more likely to influence the predictions of relation classifiers, with approximately

[3]We did not use examples with connectives to train classifiers because models trained in this way rely heavily on connectives for prediction (Pitler and Nenkova, 2009). By contrast, classifiers trained on arguments (no connectives) make predictions grounded in the semantics of examples.



(a) **Ex**plicit **w/o** Connectives (b) **Ex**plicit **w/** Connectives

(c) **Im**plicit **w/o** Connectives (d) **Im**plicit **w/** Connectives

Figure 4: Visualization of examples in PDTB 2.0 when containing or not containing a connective.

30% of the examples being predicted as different relations when containing and not containing a connective. By contrast, only about 5% of instances in the implicit corpora are predicted in different relations under the same settings.

We further visualize the representations of examples with and without a connective (see $\mathbf{v1}$ and $\mathbf{v2}$ in Algorithm 1). Figure 4 shows the visualized results on the training set of PDTB 2.0 (top-level relation) using t-SNE (van der Maaten and Hinton, 2008). Without connectives (see Fig 4a), explicit examples are well separated since the classifier is trained on arguments-label pairs. When inserting explicit connectives into inputs (see Fig 4b), the representations undergo significant changes, intertwining examples of different relations. Compared to the explicit cases, the representations of implicit instances generally remain unchanged after incorporating connectives (see Fig 4c and 4d), suggesting relations expressed by implicit arguments are barely affected by connectives.

The above results indicate that, after removing connectives, many examples in the explicit corpus express relations that differ from the original annotation. Consequently, classifiers trained on explicit examples (with connectives removed) learn a chaotic pattern for relation prediction, resulting in poor performance in real implicit scenarios.

4.4 Can label shift be measured?

Different explicit instances exhibit varying degrees of label shift. For example, the case (i) in Section 4.2 is more severe than case (ii) as deleting the connective makes the former convey a completely different relation (*Temporal* → *Contingency*) while rendering the latter ambiguous (but the original relation holds). We design a **label shift metric** to quantify the degree of label shift that occurs in each instance of an explicit corpus. We show in Sections 4.5 and 5.1 that this metric can be used to analyze factors causing label shift and to filter out noisy examples that suffer label shift, respectively.

Given an explicit corpus with annotated relations $\{(\mathrm{Arg1_i}, \mathrm{Conn_i}, \mathrm{Arg2_i}, \mathrm{Rel_i})\}|_{\mathrm{i=1}}^{\mathrm{N}}$, we first train a classifier with arguments-relation pairs. Then, for each example, we extract representations of that example when containing and not containing a connective from the trained classifier's encoder, and calculate the cosine similarity between these two representations (see value in Algorithm 1). If the cosine similarity is close to 1, it indicates that the example with and without connectives are semantically similar, thus suggesting the connective is more likely removable; otherwise, removing the connective probably results in a label shift. We compute the label shift metric for explicit corpora of PDTB 2.0 and 3.0, and find that around 33% of explicit examples in PDTB 2.0 and about 29.6% of those in PDTB 3.0 have a cosine similarity of less than 0.5, suggesting a substantial portion of connectives in the explicit dataset are not removable.

4.5 Why does label shift happen?

While we have demonstrated that label shift occurs during the construction of the *implicit-like* corpus, we know little about why removing a connective has such a significant impact. We investigate four factors that can contribute to label shift: (i) Is the removed connective a conjunction or an adverb (Prasad et al., 2006a)? Conjunctions join clauses of equal grammatical rank in a sentence or join a subordinate clause to a main clause (Blühdorn, 2008). Removing conjunctions disrupts the syntactic structure of the texts and may make the relations expressed unclear. (ii) Is the removed connective ambiguous (Webber et al., 2019a)? Some connectives, such as *since*, are ambiguous and signal multiply relations, which may result in the annotated relations of explicit examples being different from relations inferred from the arguments of these examples. (iii) Is the status of the arguments intra- or inter-sentential (Prasad et al., 2018)? The information carried by intra-sentential arguments

	PDTB 2.0		PDTB 3.0	
	coefficient	p-value	coefficient	p-value
Conj vs. Adv	-0.3946	<0.001	-0.3226	<0.001
Ambiguity	-0.0981	<0.001	-0.0412	<0.001
Intra vs. Inter	-0.1947	<0.001	-0.1898	<0.001
Input length	0.1416	<0.001	0.1944	<0.001

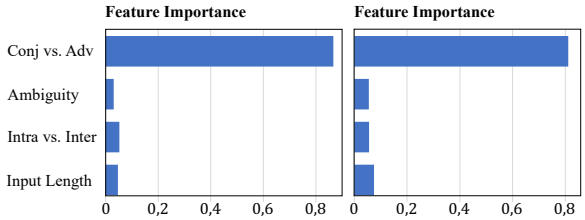
Table 1: Pearson correlation between each individual factor and the label shift metric.

is incomplete (only parts of a sentence) and may not indicate a clear relation without the help of connectives. (iv) What is the length of the input arguments? Sufficient information is the key to inferring relations from text. If the arguments are very short, it will be hard to infer a relation in the absence of connectives. We extract these four features for each example in the explicit corpus, where the first three are represented as Boolean values (i.e., 0 or 1), and the last is represented as a floating value (normalize the length to a value between 0 and 1).

We calculate the Pearson correlation between each individual factor and the label shift metric calculated in Section 4.4, and show the results on PDTB 2.0 and 3.0 (top-level relation) in Table 1. All factors are significantly correlated with the label shift metric (p-value < 0.001) but with different correlation coefficients. The syntactic role played by connectives receives the largest value (in terms of absolute value), indicating whether the removed connective is a conjunction or an adverb has the most impact on the occurrence of label shift. It is followed by the status and length of arguments. Surprisingly, ambiguity of connectives has the lowest correlation coefficient and shows a clear gap with the other factors. This suggests that ambiguity of connectives seems not to be the primary cause of label shift in PDTB 2.0 and PDTB 3.0.

The results above only show the correlation of standalone factors with label shift, without considering all factors together. Inspired by Liu et al. (2023c), we train an XGBoost model (Chen and Guestrin, 2016) to find out the importance of factors when using the four features to predict the calculated label shift metric. XGBoost is a gradient boosting framework, where the importance of a feature can be measured by the performance gain it brings (Shang et al., 2019). The framework also harnesses arbitrary interactions between features and is highly regularized to prevent overfitting, making it suitable to analyze a set of features.

We conduct the experiments on PDTB 2.0 and 3.0, and show the results in Figure 5. Consistent with the Pearson correlation analysis, the syntactic role played by connectives shows overwhelming importance in predicting the label shift metric, with an importance score of more than 0.8. In contrast, the state and length of arguments are less important when all factors are considered together. This may be because the three factors, the syntactic role played by the connective, the state of the arguments, and the length of the arguments, are not independent of each other,[4] so the latter two factors provide limited additional information to the first feature in predicting label shift. The last feature, the ambiguity of connective, is still useful but less important than other three factors. We provide more detailed setups and XGBoost results with different combination of features in Appendix B.3.



(a) PDTB 2.0 (b) PDTB 3.0

Figure 5: Feature Importance of the XGBoost Model in predicting the label shift metric on PDTB 2.0 and 3.0.

5 Strategies to alleviate the label shift

In this section, we introduce two strategies to alleviate the impact of label shift in the task of explicit to implicit relation recognition.

5.1 Filter out noisy examples

Our first strategy is straightforward: filtering out examples that may have suffered label shift. Given an explicit corpus, we calculate the cosine value of each example following the approach in Section 4.4, and filter out instances with low values. However, rather than configuring a fixed threshold for the entire corpus, we compute different thresholds per relation class. This is because data with different relations suffers from varying degrees of label shift. We group all cosine values based on the relation of each example, and calculate the averaged value for each group. If the cosine value of an instance is less than the average value of the group it belongs to, we filter it out.

[4]For example, 62.87% of explicit examples (in PDTB 2.0) whose connectives are conjunctions, contain intra-sentential arguments. And inter-sentential arguments are usually longer and contain more words than their intra-sentential counterpart.

	PDTB 2.0				PDTB 3.0			
	top-level		second-level		top-level		second-level	
Models	Acc	F1	Acc	F1	Acc	F1	Acc	F1
I2I-Entire	$67.97_{0.64}$	$59.74_{0.94}$	$58.11_{0.63}$	$37.74_{0.31}$	$72.40_{0.21}$	$67.20_{0.34}$	$62.62_{0.87}$	$53.11_{0.58}$
I2I-Reduced	$63.77_{0.53}$	$54.66_{1.31}$	$54.07_{0.83}$	$35.49_{0.49}$	$69.86_{0.91}$	$64.12_{1.29}$	$59.43_{0.40}$	$46.65_{0.83}$
Ji et al. (2015)	-	38.62	-	-	-	-	-	-
Huang and Li (2019)	-	40.90	-	-	-	-	-	-
Kurfalı and Östling (2021)	-	33.55	25.32	13.01	-	-	-	-
Common	53.73	17.48	25.22	03.66	43.62	15.19	27.69	03.10
E2I-Entire	$56.14_{0.65}$	$41.49_{0.59}$	$34.57_{0.38}$	$22.03_{0.58}$	$51.49_{0.39}$	$45.25_{0.50}$	$39.09_{0.87}$	$33.56_{0.72}$
E2I-Reduced	$55.58_{0.59}$	$39.13_{1.05}$	$31.65_{0.99}$	$18.03_{1.09}$	$48.57_{0.30}$	$40.09_{0.97}$	$36.54_{0.55}$	$28.32_{1.03}$
Our Method	$60.50_{0.34}$	$51.25_{0.70}$	$39.33_{0.28}$	$27.13_{0.50}$	$57.54_{0.16}$	$51.01_{0.45}$	$41.50_{0.30}$	$37.08_{0.13}$
w/o filtering	$58.70_{0.24}$	$45.39_{0.63}$	$36.28_{0.27}$	$23.55_{0.53}$	$52.24_{0.32}$	$46.03_{0.67}$	$40.45_{0.33}$	$34.15_{0.48}$
w/o joint learning	$57.74_{0.45}$	$44.42_{0.83}$	$35.23_{0.34}$	$22.50_{0.48}$	$52.31_{0.38}$	$44.46_{0.56}$	$40.11_{0.28}$	$33.93_{0.30}$

Table 2: Results on PDTB 2.0 and PDTB 3.0 (with standard deviation). E2I-Entire is the typical setting for explicit to implicit discourse relation recognition, serving as the baseline, and I2I-Entire is the upper bound for implicit relation classification. Our two strategies can effectively close the gap between baseline and upper bound.

5.2 Joint learning with connectives

We further investigate a joint learning framework to alleviate label shift in cases where the filtering result is imperfect. The main idea is that label shift is caused by removing connectives; so if we attempt to recover the discarded connective during training, examples may be more consistent with the original relation labels.

Given an explicit example $(\text{Arg1}, \text{Conn}, \text{Arg2}, \text{Rel})$, we insert a $\langle\text{mask}\rangle$ token between two arguments, and train a connective classifier to recover a suitable connective (conn_pred) at the masked position. Simultaneously, we train a relation classifier to predict a relation based on both input arguments and the predicted connective, i.e., $(\text{Arg1}, \text{conn\_pred}, \text{Arg2})$. With the presence of the predicted connective, we hypothesize that the modified input will be closer to the original relation than the former input containing only two arguments, alleviating the occurrence of label shift. We provide a detailed description and implementation of our method in Appendices C and D, respectively.

6 Experiments

We conduct experiments to show that our method not only improve the performance of explicit to implicit relation recognition on PDTB 2.0 and 3.0, but also works well on a corpus annotated with RST relations.

6.1 Baselines and upper bounds

We evaluate our method on PDTB 2.0 (Prasad et al., 2008) and PDTB 3.0 (Webber et al., 2019b), report the mean performance of 5 runs (with different seeds), and compare our method with existing state-of-the-art models on explicit to implicit relation recognition. In addition, we show the performance of several strong baselines and upper bounds:

- **Common**. Always predict the most common label in the training set.
- **E2I-Entire**. Finetune RoBERTa on the entire training set of explicit examples and test on implicit examples. This is the typical setting for explicit to implicit relation recognition.
- **E2I-Reduced**. Similar to E2I-Entire, but the training set is reduced to have the same size as our filtered corpus.
- **I2I-Entire**. This serves as an upper bound, where RoBERTa is finetuned on the entire training set of the implicit examples.
- **I2I-Reduced**. A variant of I2I-Entire, where the training set contains the same number of examples as our filtered corpus.

6.2 Overall results

The evaluation results on PDTB 2.0 and PDTB 3.0 are shown in Table 2. Classifiers trained on explicit corpora (E2I) perform much worse on implicit relation recognition than those trained on implicit datasets (I2I). For example, on the top-level relations in PDTB 2.0 and PDTB 3.0, E2I-Entire lags behind I2I-Entire by 18.25% and 21.95% in terms of F1 score, respectively. This is in line with previous findings that classifiers trained on explicit examples perform poorly on real implicit relations (Lin et al., 2009). Our method can substantially enhance the performance of explicit to implicit relation recognition, closing the F1 gap

with I2I-Entire from 18.25% to 8.49% in PDTB 2.0 and from 21.95% to 16.19% in PDTB 3.0. These results highlight the effectiveness of our approach for the task, which in turn suggests that label shift is one cause for poor transfer performance from explicit to implicit relations. Despite achieving impressive results, our model still has a lower performance than the upper bound (i.e., I2I-Entire). We suspect this is because even despite our suggested approach for filtering out training examples potentially affected by label shift, there may still be issues due to the fact that (1) explicit and implicit examples have different syntactic structure (Lin et al., 2009); (2) the label distributions in explicit and implicit corpora are very different (see Figure 4). These differences may cause other shifts during the transfer from explicit to implicit relation recognition, which we leave for future work.

We then analyze the effectiveness of each module in our method. Specifically, we perform an ablation study in which we systematically remove the filtering strategy, leaving our model trained with only the joint learning strategy. As shown in Table 2, removing the filtering strategy hurts the performance, with F1 scores for top-level relation recognition dropping by 5.86% and 4.98% for PDTB 2.0 and 3.0, respectively. Similarly, we eliminate the joint learning from our approach, giving it the same structure as the baseline but training it on the filtered corpus. Without jointly learning with connectives, the performance of our approach degrades (see "w/o joint learning" in Table 2), similar to the case of the filtering strategy. These results demonstrate that both strategies are crucial for achieving good performance. We also find that using each strategy individually only slightly enhances performance, below the effectiveness of combining them. This suggests that: (1) neither strategy can fully mitigate the impact of label shift; (2) the two strategies are complementary to each other since combining them achieves more improvement.

Furthermore, we analyze whether our filtering strategy can really improve data quality. To this end, we compare the performance of models trained on the same number of examples but from three different sources: our filtered corpus ("w/o joint learning"), sampling from original explicit corpus (E2I-Reduced), and sampling from implicit corpus (I2I-Reduced). Table 2 also shows the results. We find that models trained on our filtered corpus perform better than E2I-Reduced, closing the gap with I2I-Reduced. This indicates that the quality of our filtered corpus is better than the equally sized corpus obtained through random sampling.

Models	Acc	F1
I2I-Entire	$62.08_{0.41}$	$56.81_{0.72}$
I2I-Reduced	$52.87_{0.67}$	$46.51_{1.24}$
Common	35.82	07.54
E2I-Entire	$37.80_{0.76}$	$32.52_{1.34}$
E2I-Reduced	$36.26_{0.87}$	$31.28_{1.45}$
Our Method	$41.88_{0.63}$	$37.56_{0.92}$
w/o filtering	$39.90_{0.54}$	$34.15_{0.97}$
w/o joint learning	$39.04_{0.72}$	$34.75_{1.21}$

Table 3: Results on the RST GUM corpus.

6.3 Results on the GUM dataset

Our approach is based on the analysis of PDTB corpora. To test the generalizability of our approach, we evaluate it on the GUM dataset (Zeldes, 2017), which is annotated with RST relations. There are different versions of the GUM dataset, in this work we use the v9 version released by the DISRPT 2023 shared task (Braud et al., 2023). However, the GUM dataset does not have a data split of explicit and implicit relations. To address this issue, we employed a rule to divide explicit and implicit examples: if an instance (1) contains two adjacent text units and (2) contains a connective at the beginning of its second text unit, it is identified as an explicit case; otherwise, it is implicit. We provide a more detailed description of the corpus in Appendix A.1.

Table 3 shows the results of explicit to implicit relation recognition on the GUM dataset. The classifier trained on explicit examples of the GUM dataset (E2I-Entire) performs poorly in implicit relations, lagging behind its counterpart trained on implicit instances (I2I-Entire) more than 24% in F1 score. Each of our proposed strategies can slightly improve the performance on this corpus, and combining them achieves the best results, with a 5-point improvement in the F1 score over the E2I-Entire baseline. This demonstrates that our approach may generalize to other discourse data sets.

7 Conclusion

We find that one cause for the poor transfer performance from explicit to implicit relations is the occurrence of label shift when deleting connectives from explicit examples. We present both manual and empirical evidence to demonstrate the existence of such shift in the explicit corpus. We de-

sign a cosine similarity-based metric to measure label shift in the corpus, filter out noisy data, and investigate a joint learning framework to mitigate label shift. Experiments on PDTB 2.0 and PDTB 3.0 demonstrate that training classifiers on the filtered corpus with our joint learning strategy can significantly enhance the performance of explicit to implicit relation recognition. Furthermore, we show that our approach also works well on the GUM dataset, suggesting its generalizability.

8 Limitations

In this study, we conduct experiments solely on corpora annotated with PDTB and RST relations. It would be interesting to explore whether our approach is applicable to corpora annotated with other relations, such as relations in Segmented Discourse Representation Theory (SDRT, Asher et al., 2003). In addition, this work only focuses on English relational corpora. Recently, an increasing number of works have begun to call for attention to multilingual discourse (Kurfalı and Östling, 2019; Varachkina and Pannach, 2021; Liu et al., 2023a; Metheniti et al., 2023), and shared tasks (Zeldes et al., 2021; Braud et al., 2023) have been organized to deal with multilingual discourse relation classification. Therefore, investigating whether the same findings hold for discourse treebanks in other languages would be an exciting direction for research.

Acknowledgements

The authors would like to thank the four anonymous reviewers for their comments. We also thank Xiyan Fu for her valuable feedback on earlier drafts of this paper. This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany.

References

Nicholas Asher, Nicholas Michael Asher, and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

Sasha Blair-Goldensohn, Kathleen McKeown, and Owen Rambow. 2007. Building and refining rhetorical-semantic relation models. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 428–435, Rochester, New York. Association for Computational Linguistics.

Hardarik Blühdorn. 2008. Subordination and coordination in syntax semantics and discourse: Evidence from the study of connectives.

Chloé Braud, Yang Janet Liu, Eleni Metheniti, Philippe Muller, Laura Rivière, Attapol Rutherford, and Amir Zeldes. 2023. The DISRPT 2023 shared task on elementary discourse unit segmentation, connective detection, and relation classification. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 1–21, Toronto, Canada. The Association for Computational Linguistics.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Matt Ellis. 2023. How to use conjunctive adverbs. *Grammarly.com*.

David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2003. English gigaword. *Linguistic Data Consortium, Philadelphia*, 4(1):34.

Hsin-Ping Huang and Junyi Jessy Li. 2019. Unsupervised adversarial domain adaptation for implicit discourse relation classification. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 686–695, Hong Kong, China. Association for Computational Linguistics.

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

Yangfeng Ji, Gongbo Zhang, and Jacob Eisenstein. 2015. Closing the gap: Domain adaptation from explicit to implicit discourse relations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2219–2224, Lisbon, Portugal. Association for Computational Linguistics.

Congcong Jiang, Tieyun Qian, Zhuang Chen, Kejian Tang, Shaohui Zhan, and Tao Zhan. 2021. Generating pseudo connectives with mlms for implicit discourse relation recognition. In *PRICAI 2021: Trends in Artificial Intelligence*, pages 113–126, Cham. Springer International Publishing.

Najoung Kim, Song Feng, Chulaka Gunasekara, and Luis Lastras. 2020. Implicit discourse relation classification: We need to talk about evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5404–

5414, Online. Association for Computational Linguistics.

Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. 2020. Adapting BERT to implicit discourse relation classification with a focus on discourse connectives. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1152–1158, Marseille, France. European Language Resources Association.

Murathan Kurfalı and Robert Östling. 2019. Zero-shot transfer for implicit discourse relation classification. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 226–231, Stockholm, Sweden. Association for Computational Linguistics.

Murathan Kurfalı and Robert Östling. 2021. Let's be explicit about that: Distant supervision for implicit discourse relation classification via connective prediction. In *Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language*, pages 1–10, Online. Association for Computational Linguistics.

Mirella Lapata and Alex Lascarides. 2004. Inferring sentence-internal temporal relations. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 153–160, Boston, Massachusetts, USA. Association for Computational Linguistics.

Sujian Li, Liang Wang, Ziqiang Cao, and Wenjie Li. 2014. Text-level discourse dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25–35, Baltimore, Maryland. Association for Computational Linguistics.

Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. 2009. Recognizing implicit discourse relations in the Penn Discourse Treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 343–351, Singapore. Association for Computational Linguistics.

Wei Liu, Yi Fan, and Michael Strube. 2023a. HITS at DISRPT 2023: Discourse segmentation, connective detection, and relation classification. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 43–49, Toronto, Canada. The Association for Computational Linguistics.

Wei Liu, Xiyan Fu, and Michael Strube. 2023b. Modeling structural similarities between documents for coherence assessment with graph convolutional networks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7792–7808, Toronto, Canada. Association for Computational Linguistics.

Wei Liu and Michael Strube. 2023. Annotation-inspired implicit discourse relation classification with auxiliary discourse connective generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15696–15712, Toronto, Canada. Association for Computational Linguistics.

Yang Janet Liu, Tatsuya Aoyama, and Amir Zeldes. 2023c. What's hard in English RST parsing? predictive models for error analysis. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–42, Prague, Czechia. Association for Computational Linguistics.

Wanqiu Long and Bonnie Webber. 2022. Facilitating contrastive learning of discourse relational senses by exploiting the hierarchy of sense relations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10704–10716, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

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

Eleni Metheniti, Chloé Braud, Philippe Muller, and Laura Rivière. 2023. DisCut and DiscReT: MELODI at DISRPT 2023. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 29–42, Toronto, Canada. The Association for Computational Linguistics.

Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 13–16, Suntec, Singapore. Association for Computational Linguistics.

R. Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind K. Joshi, Livio Robaldo, and Bonnie Lynn Webber. 2006a. The penn discourse treebank 2.0 annotation manual.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Rashmi Prasad, Eleni Miltsakaki, Nikhil Dinesh, Alan Lee, Aravind Joshi, and Bonnie Webber. 2006b. The penn discourse treebank 1.0 annotation manual.

Rashmi Prasad, Bonnie Webber, and Alan Lee. 2018. Discourse annotation in the PDTB: The next generation. In *Proceedings 14th Joint ACL - ISO Workshop on Interoperable Semantic Annotation*, pages 87–97, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1006–1017, Vancouver, Canada. Association for Computational Linguistics.

Manami Saito, Kazuhide Yamamoto, and Satoshi Sekine. 2006. Using phrasal patterns to identify discourse relations. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 133–136, New York City, USA. Association for Computational Linguistics.

Erbo Shang, Xiaohua Liu, Hailong Wang, Yangfeng Rong, and Yuerong Liu. 2019. Research on the application of artificial intelligence and distributed parallel computing in archives classification. In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, pages 1267–1271.

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

C.E. Sporleder and A. Lascarides. 2005. Exploiting linguistic cues to classify rhetorical relations. In *Proceedings of Recent Advances in Natural Language Processing (RANLP-05)*, pages 532–539. Unknown Publisher. Pagination: 8.

Catherine Traffis. 2021. What are conjunctions? definition and examples. *Grammarly.com*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605.

Hanna Varachkina and Franziska Pannach. 2021. A unified approach to discourse relation classification in nine languages. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 46–50, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Bonnie Webber, Rashmi Prasad, and Alan Lee. 2019a. Ambiguity in explicit discourse connectives. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 134–141, Gothenburg, Sweden. Association for Computational Linguistics.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019b. The penn discourse treebank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*, 35:108.

Wei Xiang, Chao Liang, and Bang Wang. 2023. TEPrompt: Task enlightenment prompt learning for implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12403–12414, Toronto, Canada. Association for Computational Linguistics.

Amir Zeldes. 2017. The GUM corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Amir Zeldes, Yang Janet Liu, Mikel Iruskieta, Philippe Muller, Chloé Braud, and Sonia Badene. 2021. The DISRPT 2021 shared task on elementary discourse unit segmentation, connective detection, and relation classification. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 1–12, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hao Zhou, Man Lan, Yuanbin Wu, Yuefeng Chen, and Meirong Ma. 2022. Prompt-based connective prediction method for fine-grained implicit discourse relation recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3848–3858, Abu Dhabi, United Arab Emirates. 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 *Coling 2010: Posters*, pages 1507–1514, Beijing, China. Coling 2010 Organizing Committee.

PDTB 2.0	PDTB 3.0
Comparison	Comparison
Contingency	Contingency
Expansion	Expansion
Temporal	Temporal
Comparison.Concession	Comparison.Concession
Comparison.Contrast	Comparison.Contrast
Contingency.Cause	Contingency.Cause
Contingency.Pragmatic cause	Contingency.Cause+Belief
Expansion.Conjunction	Contingency.Condition
Expansion.Instantiation	Contingency.Purpose
Expansion.Alternative	Expansion.Conjunction
Expansion.List	Expansion.Equivalence
Expansion.Restatement	Expansion.Instantiation
Temporal.Asynchronous	Expansion.Level-of-detail
Temporal.Synchrony	Expansion.Manner
	Expansion.Substitution
	Temporal.Asynchronous
	Temporal.Synchronous

Table 4: Top- and second-level relations of PDTB 2.0 and PDTB 3.0 (commonly used in the literature).

Joint	Adversative	Context	Causal
Elaboration	Explanation	Contingency	

Table 5: Relations of the GUM dataset used in this work.

A Experimental settings

A.1 Dataset description

PDTB. The dataset used for analysis in this study is the Penn Discourse Treebank (PDTB). PDTB has two widely used versions, namely PDTB 2.0 (Prasad et al., 2008) and PDTB 3.0 (Webber et al., 2019b). In both versions, each example is annotated with a three-level relation from coarse to fine. In this study, we use top-level and second-level relations for analyses and experiments. Following previous work (Kim et al., 2020), we use 4 and 11 labels for top- and second-level relations in PDTB 2.0, and 4 and 14 labels for those in PDTB 3.0 (see Table 4). The two datasets are divided into training, development, and test sets, following the setup in Ji and Eisenstein (2015). Table 6 shows the statistics information on both datasets. We also show an example of the annotated connective in the implicit corpus:

(8) [He has not changed, but those around him have.]$_{\text{Arg1}}$ (**Implicit=Because**) [Many of his view on the protection of wilderness areas are now embraced by mainstream.]$_{\text{Arg2}}$

—— *Contingency.Cause*

GUM. The GUM dataset used in this study is from DISRPT 2023 (Braud et al., 2023). The original GUM dataset is annotated with a constituent tree structure, containing both structure and relation information. In DISRPT 2023, the annotated relations in the GUM v9 dataset have been converted into a group of triplets which contain text units 1 and 2, and label. Specifically, the organizers of DISRPT 2023 first convert[5] the RST constituent trees into a dependency representation (Li et al., 2014). They then extract (EDU_i, EDU_i, Relation) from the dependency tree as (text_unit1, text_unit2, label)[6] in the shared task. However, there is no split between explicit and implicit relations in this corpus. To address this issue, we follow previous research practice on RST corpus (Marcu and Echihabi, 2002) to consider examples that have two adjacent text units and contain a connective at the begining of the second text unit as explicit instances. The connective list used during the division comes from the explicit corpus of PDTB 2.0, which includes 99 distinct connectives. The processed corpus contains about 3k explicit examples and 19.2k implicit instances. Due to the small size of the explicit corpus and the uneven distribution of labels (e.g., the relation *joint* accounts for 34.8% of the explicit corpus), we only consider relations with frequency more than 100, resulting in 7 relations (see Table 5) in the final dataset. We show the statistics of the final dataset in Table 6.

Dataset	Type	Train	Dev	Test
PDTB 2.0	Explicit	14117	1462	1285
	Implicit	12632	1183	1046
PDTB 3.0	Explicit	18626	1944	1767
	Implicit	17085	1653	1474
GUM	Explicit	2095	276	264
	Implicit	11802	1607	1619

Table 6: Statistics of different corpus.

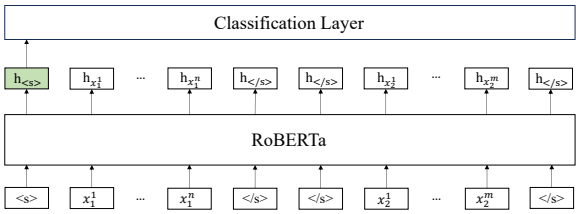


Figure 6: The architecture of the model used in the analysis.

A.2 Details in analyses

The model used in the analysis is a widely used framework for relation classification, consisting of a pre-trained model as the encoder and a linear layer as the classification layer. We follow previous work (Zhou et al., 2022; Long and Webber,

[5] https://github.com/amir-zeldes/gum

[6] We consider the text unit as argument.

Encoder Type	Relation Level	Relation Type	PDTB 2.0		PDTB 3.0	
			Same	Different	Same	Different
RoBERTa-base	Second-level	Explicit	59.21	40.79	67.56	32.44
		Implicit	93.67	6.33	93.14	6.86
RoBERTa-large	Top-level	Explicit	59.81	40.19	66.84	33.16
		Implicit	93.73	6.27	94.27	5.73
BERT-base-uncased	Top-level	Explicit	59.44	40.56	70.52	29.48
		Implicit	94.53	5.47	95.97	4.03
DeBERTa-base	Top-level	Explicit	64.64	35.36	69.41	30.59
		Implicit	96.08	3.92	94.88	5.12
T5-base	Top-level	Explicit	63.41	36.59	69.86	30.14
		Implicit	94.68	5.32	94.49	5.51

Table 7: Percentage of examples in Explicit and Implicit corpora that receive the same and different predictions when containing and not containing a connective (see diff_num in Algorithm 1).

2022) to use $\text{RoBERTa}_{\text{base}}$ as the encoder. Figure 6 shows the overall architecture of the model. We train this model following most of the default settings in RoBERTa. The optimizer used in the experiments is AdamW, with an initial learning rate of 1e-5, a batch size of 16, and a maximum epoch number of 10. The maximum input length is set to 256 for all corpora. We conduct all experiments on a single Tesla P40 GPU with 24GB memory. Note that the baselines in the experimental section (e.g., E2I-Entire, E2I-Reduced, I2I-Entire, and I2I-Reduced) use the same models as described above.

B More results about label shift

B.1 Manual analysis

We randomly sample 100 explicit examples[7] from PDTB 2.0 and remove the connective from those examples. Before starting the annotation, two students from the Computational Linguistics Department receive training in annotating relations on unannotated texts. Specifically, we introduce the definition of relations in PDTB 2.0 to the two students and ask them to practice annotating raw texts from Gigaword corpus (the corpus, like PDTB 2.0, is also in news domain). We check their annotation results, compare the difference, listen to their explanations, and give our comments. After several rounds of practice, the two students are asked to annotate these 100 explicit examples without connectives, separately. They need to annotate each instance with one of 12 relations, including *Comparison.Concession*, *Comparison.Contrast*, *Contingency.Cause*, *Contingency.Pragmatic cause*, *Expansion.Conjunction*, *Expansion.Instantiation*, *Expansion.Alternative*, *Expansion.List*, *Expansion.Restatement*, *Temporal.Asynchronous*, *Temporal.Synchrony*, and *NonRel*. The inter-annotator agreement is 0.7346 calculated in Cohen's kappa.

Algorithm 2 Identify Cases of Label Shift

Input: New and original annotations $\{(\text{NA}_\text{i}, \text{OA}_\text{i})\}|_{\text{i=1}}^{100}$
Output: Statistics of label shift: shift_num, case1, case2, case3

```
shift_num, case1, case2, case3 = 0, 0, 0, 0
for i = 1, ..., 100 do
    if NA_i ≠ OA_i then
        shift_num += 1

        if OA_i ∈ NA_i then
            case2 += 1
        else if NA_i ="NoRel" then
            case3 += 1
        else
            case1 += 1
        end if
    end if
end for
```

After completing the annotations, we implement a program (see Algorithm 2) following the definition of label shifting in equation (1), comparing the new annotations with the original ones[8]. If an example's new and original annotations are different (not exactly equal), it is considered to suffer a label shift. Further, given an example suffering a label shift, it is case (ii) if its new annotation contains the original one; otherwise, it is case (iii) if the new annotation is "no relation"; otherwise, it is case (i). Among the 100 examples, 37 are identified as suffering a label shift, in which 15, 19, and 3 belong

[7] Among the 100 examples, 16, 31, 34, and 19 are in Contingency, Comparison, Expansion, and Temporal Relations, respectively, similar to the label distribution of the explicit corpus (Contingency: 17.74%, Comparison: 29.82%, Expansion: 33.79%, Temporal: 18.65%).

[8] The original labels of these 100 explicit examples are annotated based on both argument and connective. And the inter-annotator agreement of the original explicit corpus is 0.945 (Prasad et al., 2008).

to types (i), (ii), and (iii), respectively.

B.2 Empirical evidence

Most of the analytical results in Section 4.3 are based on $\text{RoBERTa}_{\text{base}}$ and only cover top-level relations. Here, we first show the analytical results on second-level relations with $\text{RoBERTa}_{\text{base}}$. Then, we present the top-level results based on a larger size of pre-trained models, such as $\text{RoBERTa}_{\text{large}}$, or with other types of pre-trained models, such as BERT, DeBERTa, and T5. We aim to demonstrate that consistent conclusions can be drawn regardless of relation level, pre-trained model type, or scale.

Table 7 shows the results on PDTB 2.0 and PDTB 3.0. We observe similar results under different settings to Figure 3. That is, the explicit corpus has more examples receiving different predictions when containing and not containing a connective.

B.3 More XGBoost results

Setting. Given an explicit corpus with annotated relations $\{(\text{Arg1}_\text{i}, \text{Conn}_\text{i}, \text{Arg2}_\text{i}, \text{Rel}_\text{i})\}|_{\text{i=1}}^{\text{N}}$, we first calculate the label shift metric for each example as shown in Section 4.4. We then extract four features from each explicit example:

1. Is the included connective is a conjunction or an adverb? 1 for conjunction and 0 for adverb.

2. Is the included connective ambiguous? 1 for ambiguous[9] and 0 for not ambiguous.

3. Is the status of the contained arguments intra- or inter-sentential? 1 for intra-sentential and 0 for inter-sentential.

4. The number of words included in the arguments. We normalize the value between 0 and 1.

We first calculate the Pearson correlation (Liu et al., 2023b) between each individual feature and the label shift metric at the corpus level. We then train an XGBoost model using features to predict the label shift metric and analyze the importance of each feature. Table 8 shows the results when inputting different combinations of features into XGBoost. The syntactic role played by connectives contributes the most in predicting the occurrence of the label shift. When the feature of the syntactic role of connectives is not considered, the status of the arguments becomes the most important feature in predicting the label shift metric. The length of arguments and the ambiguity of connectives contribute similarly to label shift but are less important than the syntactic role played by connectives and the status of arguments.

Two Featues		Three featues	
Feature	Importance	Feature	Importance
Conj vs. Adv	0.9950	Conj vs. Adv	0.9807
Ambiguity	0.0050	Ambiguity	0.0036
Conj vs. Adv	0.9834	Intra vs. Inter	0.0156
Intra vs. Inter	0.0166	Conj vs. Adv	0.9481
Conj vs. Adv	0.9717	Ambiguity	0.0159
Input length	0.0283	Input length	0.0360
Ambiguity	0.1456	Conj vs. Adv	0.9212
Intra vs. Inter	0.8544	Intra vs. Inter	0.0374
Ambiguity	0.4543	Input length	0.0414
Input length	0.5457	Ambiguity	0.1853
Intra vs. Inter	0.8925	Intra vs. Inter	0.7058
Input length	0.1075	Input length	0.1089

Table 8: Feature importance from XGBoost when inputting different combinations of features.

Our analysis results reveal that the syntactic role played by the removed connective is an important factor in the occurrence of label shift. In PDTB, connectives come from two grammatical categories: Conjunctions (including subordinating conjunctions and coordinating conjunctions) and Adverbs (Prasad et al., 2006a). Conjunctions are used to grammatically connect clauses (Traffis, 2021), removing them can render the entire text ungrammatical and unclear in expression. For example, if we remove the *and* in "[Mr. Stein moved to a new city] and [he found a job there]", the text becomes ungrammatical and we can not know whether the text wants to express a *Conjunction* relation (with a *and*) or a *Cause* relation (with a *because*). By contrast, adverbs typically link two sentences, aiming to facilitate communication rather than serving grammatical purposes (Ellis, 2023). The elimination of adverbs may lead to reduced coherence in explicit examples, but its meaning is generally unchanged. For instance, if we remove the *however,* from "[Such problems will require considerable skill to resolve.] However, [neither Mr. Baum nor Mr. Harper has much international experience.]", the entire text becomes less coherent, but the expressed relation remains *Contrast*.

C Strategies to alleviate the label shift

C.1 Filter out noisy examples

Given an explicit relation corpus $\{(\text{Arg1}_\text{i}, \text{Conn}_\text{i}, \text{Arg2}_\text{i}, \text{Rel}_\text{i})\}|_{\text{i=1}}^{\text{N}}$, we calculate the cosine similar-

[9]We consider connectives that can signal more than one discourse relation as ambiguous. See Appendix A in Prasad et al. (2006a) and Webber et al. (2019b).

Algorithm 3 Filtering Sensitive Examples

Input: Examples with scores $\{(\mathrm{E_i}, \mathrm{Rel_i}, \mathrm{s_i})\}|_{\mathrm{i=1}}^{\mathrm{N}}$
Output: Filtered corpus C

```
groups = {}
threshold = {}
C = []
for i = 1, ..., N do
    if Rel_i in groups then
        Append(groups[Rel_i], s_i)
    else
        groups[Rel_i] = [s_i]
    end if
end for

for Rel in groups do
    threshold[Rel] = Avg(groups[Rel])
end for

for i = 1, ..., N do
    if s_i ≥ threshold[Rel_i] then
        Append(C, E_i)
    end if
end for
```

ity metric (see scores in Algorithm 1) for each example and obtain scores $\{\mathrm{s}_1, ..., \mathrm{s}_\mathrm{N}\}$. We then calculate the average scores grouped by different relations. If the cosine value of an instance is less than the average value of the group it belongs to, we filter it out (see Algorithm 3).

C.2 Joint learning with connectives

Inspired by recent work using connective information for relation classification (Kishimoto et al., 2020; Zhou et al., 2022; Liu and Strube, 2023), we investigate a joint learning framework for explicit to implicit relation recognition. Specifically, we jointly train the model to recover a connective between arguments and to predict a relation based on the recovered connective[10] and arguments. Figure 7 shows the overall architecture of the joint learning model.

Given an explicit example $(\mathrm{Arg1}, \mathrm{Conn}, \mathrm{Arg2}, \mathrm{Rel})$, where $\mathrm{Arg1} = \{\mathrm{x}_1^1, ..., \mathrm{x}_1^\mathrm{n}\}$, $\mathrm{Arg2} = \{\mathrm{x}_2^1, ..., \mathrm{x}_2^\mathrm{m}\}$, we insert an $\langle\mathrm{mask}\rangle$ token between arguments, input them to RoBERTa, and train the model to predict a connective at the masked position:

$$\mathbf{p}^c = \mathrm{softmax}(\mathbf{W}_\mathrm{c}\mathrm{h}_{\langle\mathrm{mask}\rangle} + \mathbf{b}_\mathrm{c}) \tag{2}$$

where $\mathbf{p}^\mathrm{c}$ denotes the probabilities of all connectives. We train this module with cross-entropy loss:

$$\mathcal{L}_{Conn} = -\sum_{i=0}^{N}\sum_{j=0}^{CN} C_{ij}\log(P_{ij}^c) \tag{3}$$

where C_i is the ground-truth connective of the i-th instance with a one-hot scheme, CN is the total size of connectives.

Simultaneously, we train the model to predict a relation based on both the predicted connective and arguments. To achieve so, we adopt the Gumbel-Softmax to sample a connective $\mathrm{conn\_pred}$ at the masked position:

$$\begin{aligned} g &= -\log(-\log(\xi)),\ \xi \sim \mathrm{U}(0,1) \\ \mathbf{c}_i &= \frac{\exp((\log(\mathbf{p}_i^c) + g_i)/\tau)}{\sum_j \exp((\log(\mathbf{p}_j^c) + g_j)/\tau)} \end{aligned} \tag{4}$$

where g is the Gumbel distribution, U is the uniform distribution, $\mathbf{p}_i^c$ is the probability of i-th connective output by the connective classifier, $\tau \in (0, \infty)$ is a temperature parameter (we set $\tau = 1.0$ in experiments). We use the Gumbel-Softmax rather than a normal argmax operation because the former enables the end-to-end training of the whole model, alleviating cascading errors caused by incorrectly predicted connectives. Next, we replace the $\langle\mathrm{mask}\rangle$ token with the predicted connective $\mathrm{conn\_pred}$, feed them to RoBERTa, predict a relation using the hidden states of the first token:

$$\mathbf{p}^r = \mathrm{softmax}(\mathbf{W}_r h_{\langle\mathrm{s}\rangle} + \mathbf{b}_r) \tag{5}$$

and train this module with cross-entropy loss:

$$\mathcal{L}_{Rel} = -\sum_{i=0}^{N}\sum_{j=0}^{RN} Y_{ij}\log(P_{ij}^r) \tag{6}$$

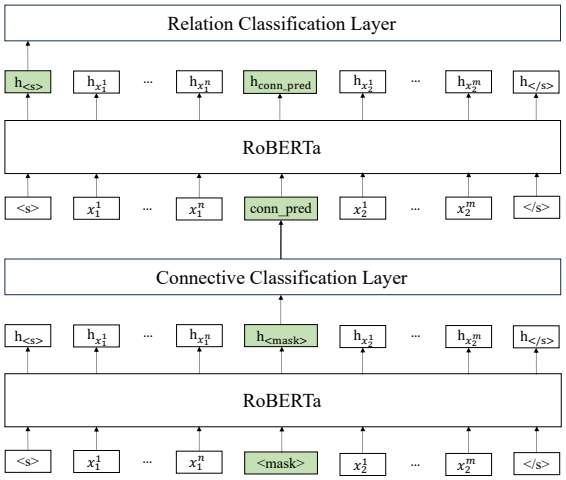


Figure 7: The architecture of the joint learning model.

[10] We can not directly input the golden connectives to the relation classifier for training. Because this will make the classifier rely heavily on golden connective for prediction (Pitler and Nenkova, 2009), which results in poor evaluation performance on implicit corpus. In the task of explicit to implicit relation recognition, golden connectives are not available during evaluation.

where Y_i is the ground-truth relation of the i-th example with a one-hot scheme, RN is the total size of relations.

We jointly train the two modules with a multi-task loss:

$$\mathcal{L} = 0.5 * \mathcal{L}_{Conn} + \mathcal{L}_{Rel} \quad (7)$$

Since our primary goal is relation prediction, we give a larger weight to relation loss $\mathcal{L}_{Rel}$.

D Implementation details

We have introduced the details of E2I and I2I baselines in Appendix A.2, here we mainly focus on the implementation of our approach.

For data filtering, we use the averaged cosine similarity score of each relation group as the threshold. This works well for PDTB 2.0 and PDTB 3.0 but we made a small adjustment to the settings for the GUM corpus. Specifically, we filter out an instance (in the GUM corpus) only if its cosine similarity score is lower than the average value of the group it belongs to and its cosine similarity score is less than 0.6. We do so because the size of the GUM corpus is small (about 2k, see Table 6). If we filter out too many instances, there will not be enough data to train classifiers to converge.

Regarding the joint learning, we use barely the same settings as baselines, including $\mathrm{RoBERTa}_{\mathrm{base}}$, AdamW optimizer, batch size of 16, learning rate of 1e-5, a maximum training epoch of 10, and maximum input length of 256.

E Discussion of threshold

In Section 5.1, we propose to use an averaging strategy to compute thresholds for different relations (called relation average). It is motivated by two observations. (1) There is a trade-off between the size and quality of the filtered corpus. If we use a large threshold, most of the noisy examples will be filtered out, but it may also filter out good examples. As a result, the filtered corpus will be relatively small, affecting the performance of the trained model (i.e., the size of the training set can affect the performance). For example, when we use 0.8 as the threshold for PDTB 2.0 (top-level relations), about 49.89% examples will be filtered out, and our full model trained on this filtered corpus can only achieve an F1-score of 46.24. If we use a small threshold, only a few noisy examples will be filtered out, but the size of the filtered corpus is close to the original corpus. In extreme cases, if we do not filter out any examples, our model will degrade into the 'w/o filtering' ablation version in Table 2. Therefore, we propose to use the average strategy to achieve a rough balance between quality and size. (2) Data with different relations suffers from varying degrees of label shift (see Fig 4a and 4b). We also tried another type of average strategy (called global average): calculate the average cosine value of all examples and use it as the threshold. On the PDTB 2.0 top-level relations, the F1 score using the global average is 48.12, lower than 51.23 using the relation average. Therefore, we chose the relation average in our paper. Tables 2 and 3 show it works well on those corpora. Since the primary goal of Sections 5 and 6 is to demonstrate that our finding of label shifting is also helpful for improving performance, we leave the exploration of better threshold selection for future work.