Revisiting Shallow Discourse Parsing in the PDTB-3: Handling Intra-sentential Implicits

Zheng Zhao and **Bonnie Webber**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh, EH8 9AB
{zheng.zhao,bonnie.webber}@ed.ac.uk

Abstract

In the PDTB-3, several thousand implicit discourse relations were newly annotated *within* individual sentences, adding to the over 15,000 implicit relations annotated *across* adjacent sentences in the PDTB-2. Given that the position of the arguments to these *intra-sentential implicits* is no longer as well-defined as with *inter-sentential implicits*, a discourse parser must identify both their location and their sense. That is the focus of the current work. The paper provides a comprehensive analysis of our results, showcasing model performance under different scenarios, pointing out limitations and noting future directions.

1 Introduction

Discourse parsing is the task of identifying and categorizing discourse relations between discourse segments in a given text. The task is considered to be important for downstream tasks such as question answering (Jansen et al., 2014), machine translation (Li et al., 2014), and text summarization (Cohan et al., 2018). There are various approaches to discourse parsing, corresponding to different views of (1) what constitutes the segments of discourse, (2) what structures can be built from such segments, and (3) what semantic and/or rhetorical relations can hold between such segments (Xue et al., 2015; Zeldes et al., 2019).

Approaches to discourse structure generally allow discourse relations to hold between segments within a sentence (i.e., *intra-sentential discourse relations*) or across sentences (i.e. *inter-sentential discourse relations*) (Joty et al., 2012; Muller et al., 2012; Stede, 2011; Stede et al., 2016).

In the Penn Discourse Treebank (PDTB; Prasad et al., 2008), all discourse relations have two arguments, called *Arg1* and **Arg2**. Discourse relations are termed *explicit*, if the evidence for the relation is an explicit discourse connective (word or phrase). For *implicit discourse relations*, evidence is in the form of argument adjacency (with or without intervening punctuation), though annotators were asked to record one or more discourse connectives that, if present, would explicitly signal the sense(s) they inferred to hold between the arguments. Where annotators felt that the relation was already signalled by an alternative (non-connective) expression, the expression was annotated as evidence for what was called an *AltLex relation* (Prasad et al., 2010).

The first major release of the PDTB was the PDTB-2 (Prasad et al., 2008) whose guidelines limited annotation to (a) Explicit relations lexicalized by discourse connectives, and (b) Implicit and AltLex relations between paragraph-internal adjacent sentences and between complete clauses within sentences separated by colons or semi-colons. Since there were only ∼530 intra-sentential implicit relations among the ∼15,500 implicit relations annotated in the PDTB-2, they were ignored in work on discourse parsing (Lin et al., 2014; Wang and Lan, 2015; Xue et al., 2015, 2016), which took implicit relations to hold only between adjacent sentences.

The situation changed with the release of the PDTB-3 (Webber et al., 2019). Among the ∼5.6K sentence-internal implicit relations annotated in the PDTB-3 are relations between VPs or clauses conjoined implicitly by punctuation (Ex. 1), between a free adjunct or free to-infinitive and its matrix clause (Ex. 2), and between a marked syntactic construction and its matrix clause. There are also implicit relations co-occurring with explicit relations (Webber et al., 2019), as noted in Section 3.1.

(1) Father McKenna moves through the house *praying in Latin*, (Implicit=and) **urging the demon to split**. [wsj_0413]

(2) *Father McKenna moves through the house* (Implicit=while) **praying in Latin**, **urging the demon to split**. [wsj_0413]

Why are implicit *intra-sentential* relations a problem for shallow discourse parsing? First, as already noted, unlike *inter-sentential* implicits, they do not occur at sentence boundaries, with material

Proceedings of the 2nd Workshop on Computational Approaches to Discourse, pages 107–121
November 10–11, 2021. ©2021 Association for Computational Linguistics

to the left of the boundary as *Arg1* and material to the right as **Arg2**. Secondly, *Arg1* and **Arg2** can appear in either order: *Arg1* before **Arg2**, as in Ex 1–2, or **Arg2** before *Arg1*, as in Ex. 3. Parsing implicit intra-sentential relations therefore requires both locating and labelling their arguments, as well as identifying the sense(s) in which they are related.[1]

(3) (Implicit=if it is) **To slow the rise in total spending**, *it will be necessary to reduce per-capita use of services*. [wsj_0314]

This work takes up some of the challenges of parsing implicit intra-sentential discourse relations. Overall, it contributes: (1) a set of BERT-based models used as a pipeline for recognizing intra-sentential implicit discourse relations as well as classifying their senses; (2) experimental evidence that these BERT-based models perform better than comparable LSTM-based models on the relation recognition task; (3) evidence that the use of parse tree features can improve model performance, as was earlier found useful in simply recognizing when a sentence contained at least one implicit intra-sentential relation (Liang et al., 2020).

2 Related Work

The focus of the current work is parsing implicit intra-sentential discourse relations in the framework of the PDTB-3. As most of the implicit relations in the PDTB-2 were inter-sentential (i.e., ∼95% of its 15.5K implicit relations), its intra-sentential implicits were ignored in parser development. Nearly all recent work on recognizing inter-sentential implicits in the PDTB-2 used neural architectures. This included multi-level attention in work by Liu and Li (2016), multiple text representations in work by Bai and Zhao (2018), including character, subword, word, sentence, and sentence pair levels, to more fully capture the text. Dai and Huang (2018) introduced a paragraph-level neural architecture with a conditional random field (CRF, Lafferty et al., 2001) layer which models inter-dependencies of discourse units and predicts a sequence of discourse relations in a paragraph. Varia et al. (2019) introduced an approach to distill knowledge from word pairs for discourse relation with CNN by joint learning of implicit and explicit relations. Shi and Demberg (2019) discovered that BERT-based models, which were trained on the next sentence prediction task, benefited implicit inter-sentential discourse relation classification. Here we assess whether they also benefit classifying intra-sentential implicit relations.

Looking at implicit relations in the PDTB-3, Prasad et al. (2017) consider the difficulty in extending implicit relations to relations that cross paragraph boundaries. Kurfalı and Östling (2019) examine whether implicit relation annotation in the PDTB-3 can be used as a basis for learning to classify implicit relations in languages that lack discourse annotation. Kim et al. (2020) explored whether the PDTB-3 could be used to learn fine-grained (Level-2) sense classification in general, while Liang et al. (2020) looked at whether separating inter-sentential implicits from intra-sentential implicits could improve their sense classification. They also took a first step towards recognizing what sentences contained intra-sentential implicit relations, finding this benefitted from the use of linearized parse tree features.

Outside the PDTB-3 framework, intra-sentential discourse relations are handled by (1) identifying discourse units (DUs), (2) attaching them to one another, and (3) associating the attachment with a coherence relation (Muller et al., 2012). One can therefore ask why we did not simply adopt this framework in the PDTB-3 and exploit the relatively good performance by systems in the DISRPT shared task on sentence-level discourse unit segmentation (Zeldes et al., 2019). There are two main reasons: First, DISRPT (and the approaches to discourse structure it covers) assumes that discourse segments cover a sentence with a non-overlapping partition. This is not the case with the PDTB, where the presence of overlapping segments (both within and across sentences) has been well documented (Lee et al., 2006). Second, discourse segments in these approaches are taken to correspond to syntactic units, which leads to both over-segmentation and under-segmentation in the PDTB-3. Of course, there are "work-arounds" for over-segmentation, such as RST's use of a SAME-SEGMENT relation (Mann and Thompson, 1988), and under-segmentation can be addressed through additional segmentation. However, we decided that starting from scratch would allow us to clearly identify the problems of parsing intra-sentential implicits, at which point, we could consider what we

[1] While the arguments to some explicit discourse relations can also appear in either order, **Arg2** is still bound to the explicit conjunction, so its relative position can be easily identified.

could adopt from work done on the DISRPT shared task on sentence-level discourse unit segmentation (Zeldes et al., 2019).

3 Methodology

Given an input sentence S represented as a sequence of tokens $s_1 \cdots s_n$, our aim is to identify the span of *Arg1* and **Arg2** if there exist an implicit discourse relation in that sentence and then to predict its corresponding sense relation. We treat the identification of argument spans as a sequence tagging problem, and the prediction of senses as a classification task. Thus, given S, our aim is to output both a tag sequence Y of length n and a sense label c for the identified relation. The generated tag sequence $y_1 \cdots y_n$ contains token-level labels where $y_j \in \{$`B-Arg1`, `B-Arg2`, `I-Arg1`, `I-Arg2`, `O`$\}$ indicating whether the token belongs to *Arg1*, **Arg2**, or Other. We adopt the BIO format (Ramshaw and Marcus, 1999) since arguments (1) can span multiple tokens, (2) can occur in either order, (3) need not be adjacent, and (4) do not overlap. (Future work will address two additional properties of intra-sentential implicits: (1) as shown in Sec 3.1, a sentence can contain more than one such relation and (2) even though most arguments are continuous spans, 264 intra-sentential implicit relations (4.2%) have discontinuous spans.)

This section describes the two parts of our approach. Section 3.1 describes the creation of two datasets based on the PDTB-3: $\mathcal{D}_1 = \{(S^{(i)}, P^{(i)}, Y^{(i)}) \mid i \in \{1 \ldots N\}\}$, $\mathcal{D}_2 = \{(A_1^{(i)}, A_2^{(i)}, P^{(i)}, c^{(i)}) \mid i \in \{1 \ldots M\}\}$ consisting of N and M input-output pairs, where S is the input sentence, P is its parse tree, A_1 and A_2 are *Arg1* and **Arg2** of the intra-sentential relation, Y is the output sentence label, and c is the sense label. Sections 3.2–3.3 describe models to recognize the argument spans and classify the relations. We provide detailed descriptions for these two steps in the rest of this section.

3.1 Dataset Generation

As we have two tasks, we built two datasets. Dataset $\mathcal{D}_1$ is used to train our argument identification models. It simply comprises individual sentences from the PDTB-3 and a sequence of labels of these sentences. Some models also contain parse tree features (Marcus et al., 1993). To generate the sequence of labels Y, we take annotations of intra-sentential implicit relations from the PDTB-3. *Arg1* tokens in the sentence are labelled `Arg1`, and **Arg2** tokens are labelled `Arg2`. Tokens with `O` labels means they belong to neither *Arg1* nor **Arg2**. If a sentence does not have any intra-sentential implicit relation, then all of its tokens will be labelled `O`. In BIO format, these labels then become `B-Arg1`, `B-Arg2`, `I-Arg1`, `I-Arg2`, and `O`.

Number of relations	Count	%
0	41,734	89.89%
1	4,314	9.29%
2	321	0.69%
3	42	0.09%
4	15	0.03%
5	4	0.01%
total	46,430	100%

Table 1: The distribution of intra-sentential implicit relations per sentence for our dataset.

The dataset comprises the 46,430 sentences in the PDTB-3, with 24,369 intra-sentential relations, of which 6,234 are implicit. Table 1 shows that a single sentence can have zero, one or more intra-sentential implicit relations, with over 99% of the sentences having no more than one. So as not to lose any training data, the 321 sentences with two relations are duplicated, with each duplicate containing one of the relations. So while we do not currently try to learn multiple implicit relations within a sentence, this approach means we don't prejudice which relation is learned.

Another way of not losing data is to treat intra-sentential AltLex relations as intra-sentential implicits because they only differ from the latter in signalling its sense in its lexicalization (Liang et al., 2020). For instance, free-adjuncts are generally **Arg2** of an intra-sentential implicit. However, those free adjuncts headed by "avoiding", "contributing to", "resulting in", etc. are labelled AltLex relations because the head uniquely signals a RESULT sense. Structurally, however, it is still an intra-sentential implicit.

Finally, the dataset does not include implicit relations that are linked to an explicit relation. Such linking is used to convey that the arguments are semantically related in a way that cannot be attributed to the explicit discourse connective alone (Webber et al., 2019). A dedicated model to recognize implicits "linked" to explicit relations is included in work by Liang et al. (2020).

Dataset $\mathcal{D}_2$ is used to train the sense classi-

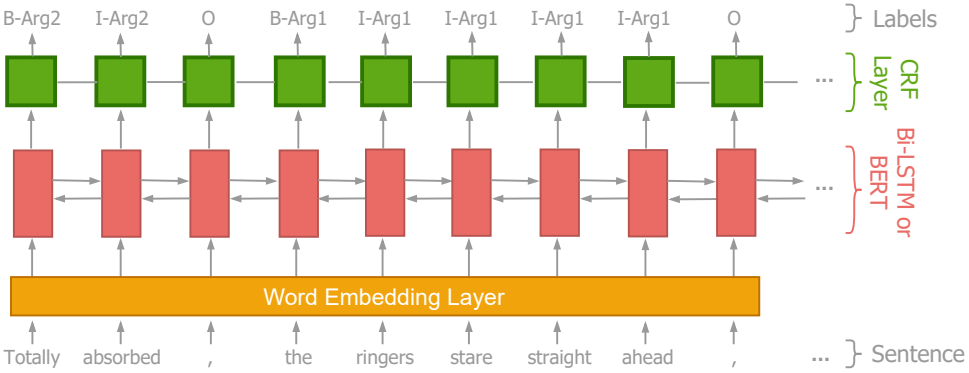

Figure 1: Architecture of our proposed argument identification model.

fier. It only contains data on sentences with intra-sentential implicits, and is thus smaller than $\mathcal{D}_1$. Each entry in $\mathcal{D}_2$ includes *Arg1*, **Arg2**, the parse tree for the sentence in which they lie, and a sense label for the *Arg1*–**Arg2** pair. Similar to $\mathcal{D}_1$, we also include AltLex relations. The current effort uses Level-2 sense labels to avoid data sparsity, while still providing a more meaningful sense than the 4 coarse labels in Level-1. The distribution of Level-2 sense labels is shown in Table 7 in Appendix A.1.

3.2 Argument Identification

The architecture for our argument identification model is shown in Figure 1. The input sentence first goes through the word embedding layer and then passes through either a BiLSTM (Hochreiter and Schmidhuber, 1997) or BERT module. Then the learned representation over the input sentence is fed into a CRF layer to generate a sequence of labels Y.

Baseline model The baseline model uses pre-trained GloVe (Pennington et al., 2014) vectors with BiLSTM and no additional parse tree features. For input sentence S where $S = \{s_1, \ldots, s_n\}$ and s_i denotes the ith token in S, the word vector e_i is obtained from the word embedding module. Then a contextualized token-level encoding h_i is obtained via a BiLSTM module:

$$\overrightarrow{h}_i = \mathbf{LSTM}_f(e_i, \overrightarrow{h}_{i-1}),$$

$$\overleftarrow{h}_i = \mathbf{LSTM}_b(e_i, \overleftarrow{h}_{i+1}),$$

$$h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i],$$

where $\overrightarrow{h}_i$ and $\overleftarrow{h}_i$ are hidden states of forward and backward LSTMs at time step i, and ; denotes concatenation.

The resulting contextual word representations are then fed to the CRF layer to predict the Y label.

BERT-based models Shi and Demberg (2019) observed that BERT-based models can benefit the task of classifying implicit *inter-sentential* discourse relations. Here we ask whether they can help in the task of argument recognition for *intra-sentential* implicit relations by creating variants of the baseline model where some parts of the model are replaced by BERT-based models.

The first variant of the model we implemented replaces the pre-trained GloVe word vectors with a pre-trained BERT model for word embedding initialization. We then construct a second variant on top of this, which also contains parse tree features. (These come from Penn TreeBank parse trees (Marcus et al., 1993), since we already know from Liang et al. (2020) that performance will drop when automated parse tree features are used.) The parse trees are first linearized and then fed to a separate BiLSTM module. The learned parse tree representations are concatenated with learned representations of the input sentence to a single vector. This vector, containing both lexical and syntactic information of the input, is then fed to the CRF layer for output prediction.

These model variants use a pre-trained BERT model. We also implemented models that fine-tune BERT on our task. One variant uses the vanilla BERT model, replacing the BiLSTM module, and another variant has the same architecture but also uses parse tree features of the input sentences.

3.3 Sense Classification

The sense classifier uses a BERT model whose input is the pair of arguments, *Arg1* and **Arg2**, and whose output is a Level-2 discourse relation sense.

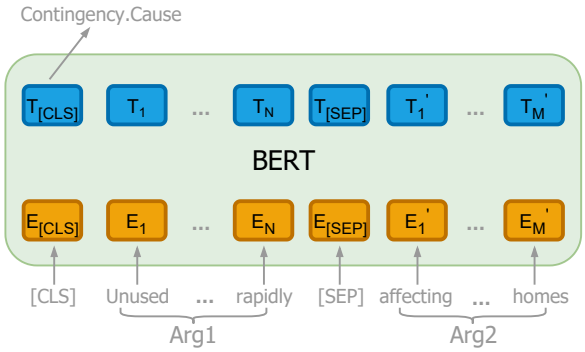

Figure 2: Architecture of our sense classifier.

The model architecture is illustrated in Figure 2. Following Shi and Demberg (2019), we concatenate *Arg1* and **Arg2** into a sequence separated with the special token [SEP]. We also insert the special token [CLS] at the beginning of the sequence. We use the output of BERT on the [CLS] token for sense classification. In addition, we also use parse trees as additional features to BERT. Similar to the method for argument identification, the learned parse tree features are concatenated with the representations of the [CLS] token to a single vector. This vector is then used for predicting the sense label.

3.4 Training and Inference

Given the training set for argument identification with labelled sequence $\{S^{(i)}, P^{(i)}, Y^{(i)} \mid i \in \{1 \dots N\}\}$, we maximize the conditional log likelihood for the sequence tagging objective:

$$\bar{w} = \underset{w}{\operatorname{argmax}} \sum_{i=1}^{N} \log p(Y^{(i)} \mid S^{(i)}, P^{(i)}, w),$$

where w denotes the model's parameters including the weights of the LSTM/BERT module and the transition weights of the CRF layer. The loss function for Y labels is the negative log-likelihood based on $Y^{(i)} = \{y_1, \dots, y_n\}$:

$$\mathcal{L}_Y = -\sum_{i=1}^{N} \sum_{j=1}^{n} \log p(y_j),$$

where $y_j \in Y^{(i)}$.

For sense classification which is predicting c label, the loss function is cross entropy:

$$\mathcal{L}_c = -\sum_{i=1}^{M} \sum_{j=1}^{C} c_j^{(i)} \log p(c_j^{(i)}),$$

where C denotes the number of classes.

At test time, inference for labels of a sentence S involves applying the Viterbi algorithm to the CRF module to find the sequence with maximum likelihood $\hat{Y}$:

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\, P(Y \mid S, P, \bar{w}).$$

4 Experiments

For our experiments with LSTM-based models, we set the hidden dimensions to 256, the word embeddings to 100, and the vocabulary size to 50K. The word embeddings are initialized either using pre-trained Glove vectors (6B tokens, *uncased*) or a pre-trained *base-uncased* BERT. For experiments with BERT-based models, we use the same configuration with *base-uncased* BERT from Devlin et al. (2019). All of our training used the Adam optimizer (Kingma and Ba, 2015). LSTM-based models used a learning rate of 1e-3 and BERT-based models used a learning rate of 5e-5. We also use gradient clipping with a maximum gradient norm of 1 and we do not use any form of regularization. For sense classification, the gradient clipping is set with a maximum gradient norm of 0.5. We carry out assessment on the datasets mentioned in Section 3.1 in two different ways: (1) using a random split of each dataset into training (60%), development (20%) and test (20%) subsets, and (2) accepting the argument in Shi and Demberg (2017) that there is too much variation across the corpus for a single random split to produce representative results and that N-fold cross-validation will deliver more reliable and predictive results. In this work, we perform 10-fold cross-validation. We use loss on the development set to perform early stopping. All of our models were trained on a single Tesla P100 GPU with a batch size of 32. Our model implementation uses PyTorch (Paszke et al., 2019). We used the BERT implementation from the Transformers library (Wolf et al., 2020), and the CRF implementation from AllenNLP library (Gardner et al., 2018) for constrained decoding for the BIO scheme.

5 Results

Argument Identification The best results here on the sequence labelling objective for all model variants come from the fine-tuned BERT-based model with additional parse tree features. In general, BERT-based models outperform LSTM-based models, and models that use parse tree features

model	*Arg1*			**Arg2**		
	Precision	Recall	F_1	Precision	Recall	F_1
LSTM + GloVe (baseline)	35.50	30.73	32.79	43.14	41.57	42.06
LSTM + BERT†	39.83	49.01	43.44	**56.93**	59.88	**58.22**
LSTM + BERT† + parse tree	43.55	50.56	46.66	53.45	**61.53**	57.19
BERT fine-tuned	45.80	50.63	48.06	54.56	59.04	56.64
BERT fine-tuned + parse tree	**49.62**∗	**51.39**	**50.43**∗	56.47	58.31	57.28

Table 2: Results of exact match for *Arg1* and **Arg2**. † denotes that a pre-trained BERT is used for word embedding, and ∗ denotes significant improvement over BERT fine-tuned with $p < 0.05$. Significance test was performed by estimating variance of the model from 5 random initialization.

model	*Arg1*			**Arg2**		
	Precision	Recall	F_1	Precision	Recall	F_1
LSTM + BERT†	44.51±6.14	51.23±5.48	47.59±4.76	58.59±2.82	62.70±2.21	60.51±1.70
LSTM + BERT† + parse tree	44.82±5.88	53.58±2.79	48.54±3.56	**60.48±2.30**	**61.64±2.68**	**61.00±1.70**
BERT fine-tuned	48.22±5.55	53.26±2.94	50.42±3.58	58.59±3.29	58.64±2.24	58.64±1.86
BERT fine-tuned + parse tree	**49.74±3.41**	**53.94±2.93**	**51.62±1.85**	60.17±2.49	60.37±2.52	60.22±1.78

Table 3: Cross-validation results of exact match for *Arg1* and **Arg2**. † denotes that a pre-trained BERT is used for word embedding.

outperform those that don't. (All results appear in Appendix A.2. The overall higher results for ○ (i.e., neither *Arg1* nor **Arg2**) come simply from label imbalance.)

The standard automatic evaluation metrics for discourse argument recognition are Precision, Recall and F_1, computed on predicted arguments for relations that match the gold annotations. We follow Xue et al. (2015) in counting an argument as correctly recognized if and only if its span exactly matches the gold argument. No reward is given for partial match. Results on the test set from the 60/20/20 random split are shown for all models in Table 2. Cross-validation results are shown in Table 3 for BERT-based models. The best overall performance comes from the fine-tuned BERT model with parse tree features. The cross-validation results also show that the use of parse tree features improves Precision, if not Recall in all cases. This makes sense as the parse tree features can be used to reject what would otherwise be False Positives. In addition, we can see that for **Arg2**, the LSTM models with pre-trained BERT perform better than fine-tuned BERT. We haven't yet figured out any reason for this. Finally, we can see that models perform better on **Arg2** (both Recall and Precision) than on *Arg1*. We attribute this to the fact that, even though **Arg2** is not marked with an explicit connective (i.e., these are all implicit relations), there may still be positional and/or syntactic cues to its identity.

Supporting these observations are statistics from the test set. First, of its 987 sentences with intra-sentential implicit relations, the leftmost argument aligns with the beginning of the sentence 495 times (50.2%). Of these, *Arg1* is the leftmost argument 430 times (86.9%). At the other end of the sentence, 685 (69.4%) relations have their rightmost argument ending at the end of the sentence, almost 200 more than those with an argument at the beginning. Of these 685 relations, 614 (89.6%) have **Arg2** at their rightmost boundary. Note that whenever an argument starts or ends at a sentence boundary, the model just needs to predict the other end of the span, which is easier than predicting both ends. With **Arg2** appearing more often at a sentence boundary than *Arg1*, this is consistent with our observation of model performance.

Even though the training data for the argument identification model contains at most one relation per sentence and the model only predicts continuous argument spans, we purposely relax the constraint for the model to predict at most one relation per sentence. We find that for all test set predictions, our best model predicts 77 sentences with more than one *Arg1* (7.53% of all sentences predicted to have a relation). It also predicts 44 sentences with more than one **Arg2** (4.31% of all sentences predicted to have a relation). This probably arises from duplicating those training instances with more than one relation (cf. Section 3.1). Later in this section, we show how the sense classifier can be used

model	*Arg1*-**Arg2**			**Arg2**-*Arg1*		
	Precision	Recall	F_1	Precision	Recall	F_1
LSTM + GloVe (baseline)	70.50	42.94	53.37	45.16	18.92	26.67
LSTM + BERT†	70.03	75.03	74.01	51.85	56.76	54.19
LSTM + BERT† + parse tree	70.82	70.97	70.90	39.22	27.03	32.00
BERT fine-tuned	**83.77**	73.49	**78.30**	59.09	70.27	64.20
BERT fine-tuned + parse tree	78.35	**76.12**	77.22	**62.22**	**75.68**	**68.29**

Table 4: Results of argument order prediction with either *Arg1*-**Arg2** (914 instances from the test set) and **Arg2**-*Arg1* (73 instances from the test set). † denotes a pre-trained BERT is used for word embedding.

to choose among multiple predicted arguments.

As noted earlier, the order of arguments in an intra-sentential implicit relation is not fixed. So we have also correlated model performance with argument order. Of the 5,157 sentences overall with intra-sentential implicit relations, 4,665 (90.5%) have arguments in *Arg1*-**Arg2** order. In the 60/20/20 split test set, this imbalance is higher across the 987 intra-sentential implicit relations, with 914 (92.6%) showing the more common *Arg1*-**Arg2** order. Performance of the model in predicting argument order is shown in Table 4. Note that a match of argument order here does not imply exact match of arguments in the previous evaluation. However, all models are more accurate in predicting the more frequent *Arg1*-**Arg2** order than **Arg2**-*Arg1* order, although the performance difference is less with BERT-based models than with LSTM-based models, suggesting BERT can deal with under-represented data better.

For completeness, we also analyze the best model's argument identification performance under different conditions. First, we consider that part of the 60/20/20 split test set whose sentences contain more than one intra-sentential implicit relations. For a fair comparison, we ensure that all single-relation tokens derived from the same sentence are in the test set (36 tokens). We further divide these tokens into ones whose relation is further left in the sentence and ones whose relation is further right. The top part of Table 5 shows the performance of the fine-tuned BERT model with parse tree features on these sets. Compared with the original test set, the overall performance on relations derived from sentences with multiple relations is much worse. This is expected as the training set for argument identification assumed that a sentence can only have at most one intra-sentential implicit relation.

We next examine performance for different sense labels, based on the four most frequent senses in the test set – those that appear more than 100 times. The bottom part of Table 5 shows that our model performed best on CONTINGENCY.CAUSE. Inspection of the True Positives shows this to result from **Arg2** of these relations often being headed by a AltLex token with Part-of-speech tag VBG, which uniquely conveys the sense CONTINGENCY.CAUSE.RESULT. The model can thus easily learn to recognize these arguments using the parse tree features.

Sense Classification Cross-validation results for Precision, Recall, and F_1 for the top four senses recognized by our sense classifier are shown in Table 6. The complete performance breakdown is given in Table 14 in Appendix A.3, along with a confusion matrix (Figure 3) and results on the test set in the 60/20/20 random split (Table 13). As the distribution of senses is imbalanced, we calculate overall performance by averaging performance for each sense, weighted by its frequency in the test set. Note that the overall F_1 for intra-sentential implicits is much higher than 50.41 reported in Liang et al. (2020) where a LSTM-based model is used. The overall accuracy of our sense classifier on cross-validation is 69.54%, and that on the test set in the 60/20/20 random split is 75.19%. From the results, we can see that the performance for CONTINGENCY.PURPOSE is very high, with F_1 exceeding the weighted average by almost 20 points. We speculate that this is due to the fact that over 90% of CONTINGENCY.PURPOSE labels are on relations where **Arg2** begins with a free "to clause" and there are few other intra-sentential implicits labelled CONTINGENCY.PURPOSE.

Our sense classifier is trained using gold argument spans taken from the PDTB-3. In reality, such gold annotations will rarely be available. Thus, we also use predicted arguments from our argument identification model as inputs to test the ability of

condition	*Arg1*			**Arg2**			# of rels
	Precision	Recall	F_1	Precision	Recall	F_1	
multiple relations	31.25	28.57	29.85	21.95	25.71	23.68	36
multiple relation + left	25.00	23.53	24.24	10.00	11.76	10.81	18
multiple relation + right	37.50	33.33	35.29	33.33	38.89	35.90	18
Contingency.Cause	**68.12**	**62.65**	**65.27**	75.44	**69.08**	**72.12**	249
Contingency.Purpose	63.98	47.22	54.34	**78.80**	57.54	66.51	252
Expansion.Conjunction	48.08	39.06	43.10	42.86	39.84	41.30	128
Expansion.Level-of-detail	59.05	56.88	57.94	60.38	58.72	59.53	109
Original test set	49.62	51.39	50.43	56.47	58.31	57.28	987

Table 5: Results of argument identification for *Arg1* and **Arg2** for our best model under different conditions.

Sense	BERT + Parse Tree		
	Precision	Recall	F_1
Contingency.Cause	76.50 ± 4.22	74.60 ± 4.89	75.31 ± 2.14
Contingency.Purpose	86.78 ± 2.05	90.49 ± 2.66	88.56 ± 1.57
Expansion.Conjunction	73.99 ± 8.05	70.33 ± 5.55	71.82 ± 5.03
Expansion.Level-of-detail	57.79 ± 4.70	48.23 ± 3.12	52.37 ± 2.06
Overall	70.86 ± 1.45	69.54 ± 1.30	69.65 ± 1.36

Table 6: Cross-validation results for sense classification.

our sense classifier. Specifically, we first obtain test set sentences with intra-sentential implicit relations and their parse trees. Then we feed those to our argument identification model to get predicted arguments. These predicted arguments are then fed in to our sense classifier together with the parse tree features. If the argument identification model fails to predict any arguments for any sentences, we ignore them in our evaluation. Table 15 in Appendix A.3 shows the sense classification results using predicted arguments and, for comparison, the results using the gold arguments. Note that as we dropped some sentences, the results with gold arguments are slightly different from those of the original test set. Because the performance drop is small going from gold arguments to predicted arguments, we can argue that our models can be used as a pipeline for handling intra-sentential implicit relations in shallow discourse parsing with the input simply being a sentence.

We noted in Section 3.1 that our argument identification model might predict multiple *Arg1*s and/or **Arg2**s for a given sentence. We therefore assessed whether the sense classifier could be used to decide which of the predicted arguments to use. Specifically, we identify cases where one argument has a single prediction but there are multiple predictions for the other argument. For each *Arg1*-**Arg2** pair, we use the sense classifier to predict the sense label and its likelihood. Comparing these likelihoods, we select the pair with the highest certainty, ignoring cases where the predicted senses are the same for all pairs. We also implement a baseline of always choosing the pair with the most frequent sense, which is CONTINGENCY.CAUSE. The 60/20/20 split test set shows 34 cases in which there are multiple predictions of *Arg1* for a single **Arg2**. In 23 out of these 34 instances, the correct *Arg1* is associated with the pair with the highest likelihood. The baseline only gets 13 cases correct. Similarly, the test set shows 23 cases in which there are multiple predictions of **Arg2** for a single *Arg1*. In 13 of these 23 instances, the pair for which the sense classifier assigns the highest priority correctly identifies which **Arg2** to use. The baseline only gets 9 cases correct. While further analysis is needed, this does show that the sense classifier can contribute to selecting the right argument from the set of predicted argument candidates.

6 Conclusions and Future Work

To the best of our knowledge, this is the first work to attempt to identify the arguments of intra-sentential implicit discourse relations in the framework of the PDTB-3, as well as their order and at least one of their sense relations. We used a model architecture similar to models of a sequence tag-

ging task, and concluded that BERT-based models have better performance than LSTM-based models in both exactly matching gold annotations of arguments and correctly predicting the order of *Arg1* and **Arg2**. We confirmed that using parse trees features as input to the model assists with these tasks. We also provide evidence that our sense classifier, together with the argument recognizer, can be used as a pipeline for handling intra-sentential implicit relations. We also find that the sense classifier can be used to aid the selection of the right argument from the set of predicted argument candidates.

Our methods have several limitations. First, we assumed every sentence can have at most one intra-sentential implicit relation, whereas in reality, multiple such relations are possible (cf. Table 1). Secondly, our approach does not support the case of discontinuous argument spans. Thirdly, we have ignored implicit relations "linked" to explicit relations (either intra-sentential or inter-sentential). Finally, although we have followed the lead of Shi and Demberg (2017) in assessing performance using cross-validation because it is more reliable than choosing a specific test set, we did not cast all our results in those terms. In the future, we plan to address these problems and develop methods that can identify all existing relations within the sentence.

Acknowledgments

This work was supported in part by the UKRI Centre for Doctoral Training in Natural Language Processing (UKRI grant EP/S022481/1), the University of Edinburgh, and Huawei. We would like to thank Hannah Rohde and anonymous CODI reviewers for their helpful feedback.

References

Hongxiao Bai and Hai Zhao. 2018. Deep enhanced representation for implicit discourse relation recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 571–583, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621, New Orleans, Louisiana. Association for Computational Linguistics.

Zeyu Dai and Ruihong Huang. 2018. Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 141–151, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew E. Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. *CoRR*, abs/1803.07640.

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

Peter Jansen, Mihai Surdeanu, and Peter Clark. 2014. Discourse complements lexical semantics for non-factoid answer reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 977–986, Baltimore, Maryland. Association for Computational Linguistics.

Shafiq Joty, Giuseppe Carenini, and Raymond Ng. 2012. A novel discriminative framework for sentence-level discourse analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 904–915, Jeju Island, Korea. Association for Computational Linguistics.

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.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

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.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann.

Alan Lee, Rashmi Prasad, Aravind Joshi, Nikhil Dinesh, and Bonnie Webber. 2006. Complexity of dependencies in discourse: Are dependencies in discourse more complex than in syntax. In *Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories*, pages 12–23.

Junyi Jessy Li, Marine Carpuat, and Ani Nenkova. 2014. Assessing the discourse factors that influence the quality of machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 283–288, Baltimore, Maryland. Association for Computational Linguistics.

Li Liang, Zheng Zhao, and Bonnie Webber. 2020. Extending implicit discourse relation recognition to the PDTB-3. In *Proceedings of the First Workshop on Computational Approaches to Discourse*, pages 135–147, Online. Association for Computational Linguistics.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

Yang Liu and Sujian Li. 2016. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1233, Austin, Texas. Association for Computational Linguistics.

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

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Philippe Muller, Stergos Afantenos, Pascal Denis, and Nicholas Asher. 2012. Constrained decoding for text-level discourse parsing. In *Proceedings of COLING 2012*, pages 1883–1900, Mumbai, India. The COLING 2012 Organizing Committee.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

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, Katherine Forbes Riley, and Alan Lee. 2017. Towards full text shallow discourse relation annotation: Experiments with cross-paragraph implicit relations in the PDTB. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 7–16, Saarbrücken, Germany. Association for Computational Linguistics.

Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2010. Realization of discourse relations by other means: Alternative lexicalizations. In *Coling 2010: Posters*, pages 1023–1031, Beijing, China. Coling 2010 Organizing Committee.

Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*, pages 157–176. Springer.

Wei Shi and Vera Demberg. 2017. On the need of cross validation for discourse relation classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 150–156, Valencia, Spain. Association for Computational Linguistics.

Wei Shi and Vera Demberg. 2019. Next sentence prediction helps implicit discourse relation classification within and across domains. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5790–5796, Hong Kong, China. Association for Computational Linguistics.

Manfred Stede. 2011. Discourse processing. *Synthesis Lectures on Human Language Technologies*, 4(3):1–165.

Manfred Stede, Stergos Afantenos, Andreas Peldszus, Nicholas Asher, and Jérémy Perret. 2016. Parallel

discourse annotations on a corpus of short texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1051–1058, Portorož, Slovenia. European Language Resources Association (ELRA).

Siddharth Varia, Christopher Hidey, and Tuhin Chakrabarty. 2019. Discourse relation prediction: Revisiting word pairs with convolutional networks. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 442–452, Stockholm, Sweden. Association for Computational Linguistics.

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

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. 2019. The penn discourse treebank 3.0 annotation manual. https://catalog.ldc.upenn.edu/docs/LDC2019T05/PDTB3-Annotation-Manual.pdf.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

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

Nianwen Xue, Hwee Tou Ng, Sameer Pradhan, Attapol Rutherford, Bonnie Webber, Chuan Wang, and Hongmin Wang. 2016. CoNLL 2016 shared task on multilingual shallow discourse parsing. In *Proceedings of the CoNLL-16 shared task*, pages 1–19, Berlin, Germany. Association for Computational Linguistics.

Amir Zeldes, Debopam Das, Erick Galani Maziero, Juliano Antonio, and Mikel Iruskieta. 2019. The DISRPT 2019 shared task on elementary discourse unit segmentation and connective detection. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 97–104, Minneapolis, MN. Association for Computational Linguistics.

A Appendix

A.1 Distribution of Sense Label

The distribution of intra-sentential implicit sense labels is shown in Table 7. We can see that the sense labels are very imbalanced.

Sense	Count	%
Comparison.Concession	66	1.28%
Comp.Concession+S.A.	4	0.08%
Comparison.Contrast	112	2.17%
Comparison.Similarity	7	0.14%
Contingency.Cause	1366	26.49%
Contingency.Cause+Belief	66	1.28%
Cont.Cause+SpeechAct	1	0.02%
Contingency.Condition	222	4.30%
Cont.Condition+SpeechAct	1	0.02%
Cont.Negative-condition	1	0.02%
Contingency.Purpose	1323	25.65%
Expansion.Conjunction	667	12.93%
Expansion.Disjunction	17	0.33%
Expansion.Equivalence	35	0.68%
Expansion.Instantiation	86	1.67%
Expansion.Level-of-detail	565	10.96%
Expansion.Manner	183	3.55%
Expansion.Substitution	82	1.59%
Temporal.Asynchronous	178	3.45%
Temporal.Synchronous	175	3.39%

Table 7: The distribution of Level-2 labels for intra-sentential implicit relations in our dataset. Comp is short for Comparison, Cont is short for Contingency, and S.A. is short for SpeechAct.

A.2 Additional Results for Argument Identification

In this section, we provide the results of the sequence tagging objective for all models variants as well as the results of argument order prediction. All results are reported on the test set. Table 8 shows results for our baseline LSTM model using GloVe word embeddings. Table 9 then shows results for the model when GloVe word embedding are replacrd with BERT embeddings. There is a large increase in Recall, with no drop in Precision except for `O` labels. Table 10 shows results when parse tree features are added to the model. Here, there is an increase in Recall for *Arg1*, and while Recall for **Arg2** decreases somewhat, it is accompanied by an increase in Precision. Then, we experiment with the vanilla BERT model and fine-tuning it on our dataset. The results are provided in Table 11. We can again observe a slight increase in performance, showcasing the power of BERT, even without the parse tree features. Finally, the last model we implement is BERT taking additional parse tree features fine-tuned on our dataset. The results are shown in Table 12. We can see that this is the best performing model for argument identification, proving that both BERT and parse tree features can improve performance.

Label	Precision	Recall	F_1
`B-Arg1`	46.50	35.62	40.43
`B-Arg2`	54.07	46.76	50.15
`I-Arg1`	54.46	46.51	50.17
`I-Arg2`	48.64	60.03	53.74
`O`	95.38	95.24	95.31

Table 8: Results of the baseline LSTM model on test set using GloVe word embedding.

Label	Precision	Recall	F_1
`B-Arg1`	48.10	52.35	50.14
`B-Arg2`	60.88	69.11	64.73
`I-Arg1`	63.34	65.59	64.45
`I-Arg2`	67.42	70.53	68.94
`O`	88.84	96.47	92.50

Table 9: Results of the LSTM model on test set using BERT word embedding.

Label	Precision	Recall	F_1
`B-Arg1`	50.29	56.77	53.33
`B-Arg2`	63.57	61.13	62.33
`I-Arg1`	59.68	69.55	64.24
`I-Arg2`	73.08	66.74	69.77
`O`	88.86	96.51	92.53

Table 10: Results of the LSTM model on test set using BERT word embedding and parse tree feature.

A.3 Additional Results for Sense Classification

In this section, we provide the full breakdown of performance of our sense classifier on each sense label. The results are provided in Table 13. Note that we only included senses existing in the test set. We provide a confusion matrix in Figure 3. In addition, we also provide results using 10-fold cross-validation in Table 14. We also provide the full comparison for sense classification results using

Label	Precision	Recall	F_1
B-Arg1	56.33	56.86	56.59
B-Arg2	65.71	64.60	65.15
I-Arg1	70.57	64.10	67.18
I-Arg2	73.90	65.96	69.70
O	88.72	97.68	92.98

Table 11: Results on test set using fine-tuning BERT model without parse tree features.

Label	Precision	Recall	F_1
B-Arg1	52.65	58.74	55.53
B-Arg2	61.27	66.10	63.60
I-Arg1	63.52	70.54	66.85
I-Arg2	69.32	71.93	70.60
O	89.10	96.46	92.64

Table 12: Results on test set using fine-tuning BERT model with parse tree features.

predicted arguments versus using gold arguments in Table 15. We can observe that for most senses, results using predicted arguments have slight inferior performance than those using gold arguments. For cases where results using predicted arguments are better, they are all under-represented senses in the dataset.

Sense	Precision	Recall	F_1
Comparison.Concession	10.0	7.14	8.33
Comparison.Contrast	40.74	84.62	55.0
Contingency.Cause	79.43	76.45	77.91
Contingency.Cause+Belief	15.0	50.0	23.08
Contingency.Condition	68.52	84.09	75.51
Contingency.Purpose	93.09	92.71	92.90
Expansion.Conjunction	88.79	76.30	82.07
Expansion.Disjunction	0.00	0.00	0.00
Expansion.Equivalence	0.00	0.00	0.00
Expansion.Instantiation	64.71	73.33	68.75
Expansion.Level-of-detail	60.95	53.78	57.14
Expansion.Manner	48.28	68.29	56.57
Expansion.Substitution	53.33	72.73	61.54
Temporal.Asynchronous	72.97	65.85	69.23
Temporal.Synchronous	66.67	76.92	71.43
Micro Average	75.19	75.19	75.19
Weighted Average	75.98	75.19	75.19

Table 13: Full results on test set for sense classification. We only included senses existing in the test set.

Sense	Precision	Recall	F_1
Comparison.Concession	21.75 ± 21.91	17.62 ± 19.07	17.15 ± 15.73
Comparison.Contrast	45.79 ± 16.14	47.23 ± 12.42	44.93 ± 12.39
Contingency.Cause	76.50 ± 4.22	74.60 ± 4.89	75.31 ± 2.14
Contingency.Cause+Belief	11.44 ± 8.82	21.78 ± 17.88	14.62 ± 11.41
Contingency.Condition	73.29 ± 9.99	77.34 ± 8.69	74.28 ± 3.60
Contingency.Purpose	86.78 ± 2.05	90.49 ± 2.66	88.56 ± 1.57
Expansion.Conjunction	73.99 ± 8.05	70.33 ± 5.55	71.82 ± 5.03
Expansion.Disjunction	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Expansion.Equivalence	24.83 ± 31.10	20.67 ± 21.33	20.67 ± 21.33
Expansion.Instantiation	25.97 ± 14.89	39.10 ± 24.73	29.11 ± 13.85
Expansion.Level-of-detail	57.79 ± 4.70	48.23 ± 3.12	52.37 ± 2.06
Expansion.Manner	44.10 ± 12.40	47.07 ± 9.91	44.18 ± 7.96
Expansion.Substitution	68.01 ± 22.64	66.02 ± 15.04	65.23 ± 17.92
Temporal.Asynchronous	57.88 ± 13.06	51.39 ± 13.06	52.81 ± 9.81
Temporal.Synchronous	56.99 ± 13.29	53.73 ± 17.83	53.84 ± 13.07
Micro Average	69.54 ± 1.30	69.54 ± 1.30	69.54 ± 1.30
Weighted Average	70.86 ± 1.45	69.54 ± 1.30	69.65 ± 1.36

Table 14: Full results on cross-validation for sense classification.

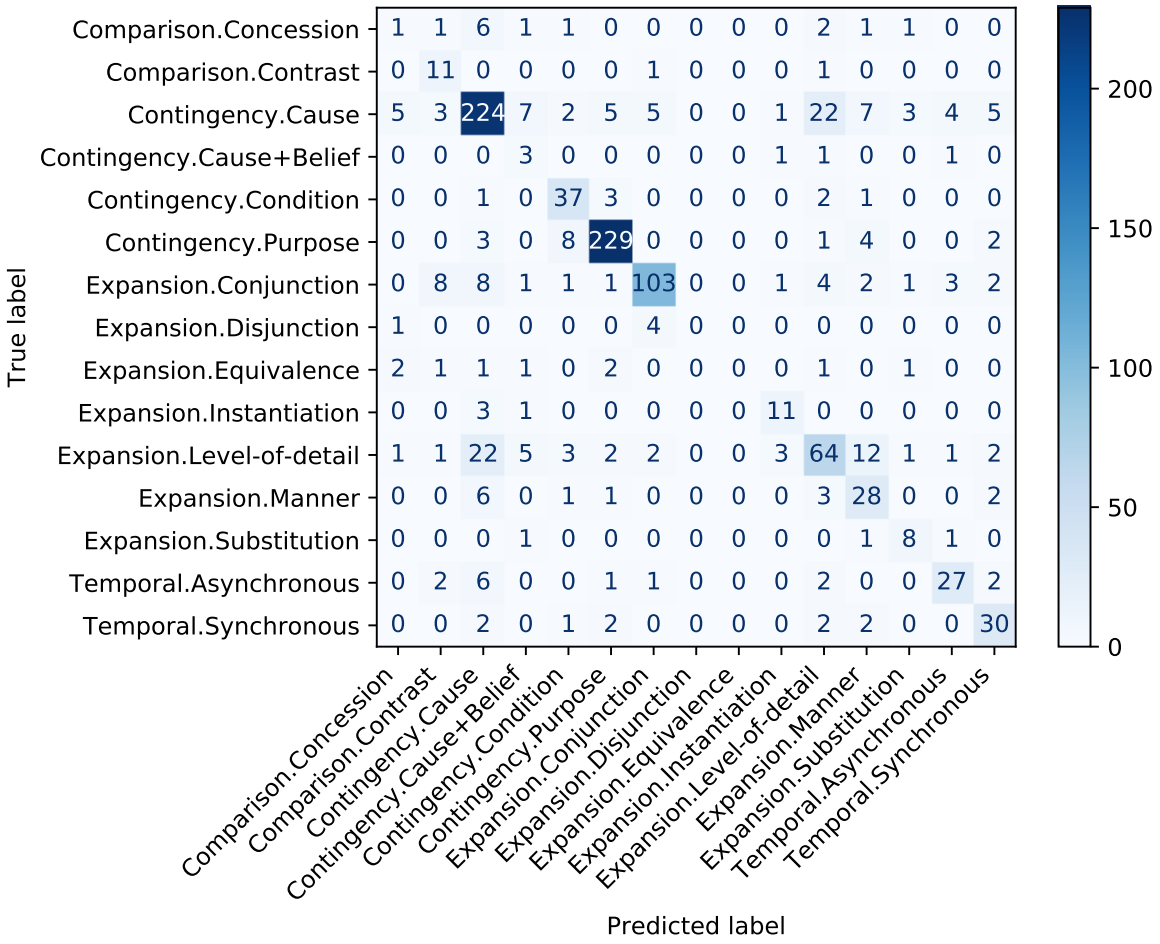

Figure 3: Confusion matrix of our sense classifier on the test set.

Sense	Predicted arguments			Gold arguments		
	Precision	Recall	F_1	Precision	Recall	F_1
Comparison.Concession	11.11	8.33	9.52	10.0	8.33	9.09
Comparison.Contrast	34.62	75.0	47.37	40.74	91.67	56.41
Contingency.Cause	75.93	73.21	74.55	79.25	75.00	77.06
Contingency.Cause+Belief	17.65	60.0	27.27	11.11	40.0	17.39
Contingency.Condition	66.04	85.37	74.47	62.96	82.93	71.58
Contingency.Purpose	85.90	88.64	87.25	87.01	91.36	89.14
Expansion.Conjunction	74.55	68.33	71.30	83.00	69.17	75.45
Expansion.Disjunction	0.00	0.00	0.00	0.00	0.00	0.00
Expansion.Equivalence	0.00	0.00	0.00	0.00	0.00	0.00
Expansion.Instantiation	44.44	33.33	38.10	58.33	58.33	58.33
Expansion.Level-of-detail	53.41	43.52	47.96	57.30	47.22	51.78
Expansion.Manner	40.00	55.56	46.51	42.31	61.11	50.0
Expansion.Substitution	57.14	72.73	64.0	50.0	72.73	59.26
Temporal.Asynchronous	62.86	56.41	59.46	66.67	56.41	61.11
Temporal.Synchronous	65.0	68.42	66.67	63.41	68.42	65.82
Micro Average	69.30	69.30	69.30	71.52	71.52	71.52
Weighted Average	69.35	69.30	68.95	72.27	71.52	71.39

Table 15: Comparison of full results on test set for sense classification using predicted arguments versus using gold arguments.