Investigating the Use of Transformer Based Embeddings for Multilingual Discourse Connective Identification

Thomas Chapados Muermans

A Thesis

in

The Department

of

Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Computer Science at

Concordia University

Montréal, Québec, Canada

June 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Thomas Chapados Muermans**

Entitled: **Investigating the Use of Transformer Based Embeddings for Multilingual Discourse Connective Identification**

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Tristan Glatard

______________________________ Examiner
Dr. Olga Ormandjieva

______________________________ Examiner
Dr. Tristan Glatard

______________________________ Supervisor
Dr. Leila Kosseim

Approved by ______________________________
Dr. Lata Narayanan, Chair
Department of Computer Science and Software Engineering

____________ 2022

____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Investigating the Use of Transformer Based Embeddings for Multilingual Discourse Connective Identification

Thomas Chapados Muermans

In this thesis, we report on our experiments toward multilingual discourse connective (or DC) identification and show how language-specific BERT models seem to be sufficient even with little task-specific training data and do not require any additional handcrafted features to achieve strong results. Although some languages are under-resourced and do not have large annotated discourse connective corpora. To address this, we developed a methodology to induce large synthetic discourse annotated corpora using a parallel word aligned corpus. We evaluated our models in 3 languages: English, Turkish, and Mandarin Chinese; and applied our induction methodology on English-Turkish and English-Chinese. All our models were evaluated in the context of the recent DISRPT 2021 Task 2 shared task. Results show that the F-measure achieved by our simple approach (93.12%, 94.42%, 87.47% for English, Turkish and Chinese) are near or at state-of-the-art for the 3 languages while being simple and not requiring any handcrafted features.

Acknowledgments

I would like to thank Dr. Leila Kosseim. She took a chance on me even though she didn't know me as a student. She was always available to help guide me in my research and always did so with kindness and patience. I don't think this would have gone this smoothly without her. Additionally I thank Dr. Tristan Glatard and Dr. Olga Ormandjieva for their valuable comments on an earlier version of this thesis.

Secondly I would like to thank Christophe Marcellin, Benoit Girouard Bond and Dominic Bouffard for writing recommendation letters. These letters were instrumental in getting accepted into the graduate program at Concordia.

I would also like to thank my colleagues in the CLaC lab Andrés Lou, Pavel Kholopin, Farhood Farahnak and Hessam Amini for always being willing to help me with the technical issues I faced throughout my research, as well as, always providing great feedback on my dry runs.

Lastly, I would like to dedicate this thesis to my partner Felix Pifalo. From the very beginning, Felix has been supporting and encouraging me even though it meant putting our life plans on hold for a little while. No matter how often my music was too loud he was always patient, loving and understanding. Working beside him every day for the past two years has made this journey not only possible but wonderful. I truly don't think I could have done this without him.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A discourse is a coherent group of sentences, as opposed to just a random assemblage of sentences (Jurafsky and Martin, 2009). A coherent text flows smoothly from beginning to end, and allows the reader to follow the logical relationships between ideas. These relationships, also known as discourse relations (DRs), can be signaled explicitly using specific words or phrases. For example, in the sentence:

(Ex. 1) *Shorter maturities are considered a sign of rising rates* because **portfolio managers can capture higher rates sooner.**

the word *because* relates the first idea (in italics) to the second (in bold) and signals a causality relation. In the context of natural language processing (NLP), these explicit words are called discourse connectives (DCs). Automatically identifying these DCs can be useful for many NLP tasks, such as question answering (Jansen et al., 2014) and text summarization (Louis et al., 2010) where it is important to measure how coherent the generated text is.

DCs come in different forms. They can be `Explicit`, as in (Ex. 1) above, when typical words or phrases, drawn from specific grammatical categories are used to signal the discourse relation (Prasad et al., 2008). Today, state of the art models for automatically identifying `Explicit` connectives in English reach human level performance[1]; however, in English, `Explicit` connectives

[1] Indeed Johannsen and Søgaard (2013) showed that a simple logistic regression model could achieve an F-measure above 94% for `Explicit` connective identification; whereas the human performance is estimated to be 82.8% (Miltsakaki et al., 2004) for the more general task of discourse parsing which includes DC identification, argument segmentation and relation labeling.

account for only 45.47% of all DCs tagged in the The Penn Discourse Treebank PDTB (Prasad et al., 2008). The other 54.53% fall under the umbrella of `Implicit` relations, which are defined as "relations between abstract objects that are not realized explicitly in the text and are left to be inferred by the reader" (Prasad et al., 2008). (Ex. 2) shows an example of an `Implicit` relation.

(Ex. 2) *Some have raised their cash positions to record levels.* **High cash positions help buffer a fund when the market falls.**

In (Ex. 2) the reader can infer a relation of causality, although it is not explicitly signaled by a DC; there is an implicit *because* between the sentences. `Implicit` relations are difficult to identify automatically because no typical lexical marker from a well-defined grammatical category directly signals them. However some `Implicit` relations are signaled by textual units that use alternative lexicalizations (`AltLex`) to `Explicit` DCs. `AltLex` are an open class of phrases present in sentences where providing an `Explicit` DC would lead to a redundancy in the expression of the relation. Alternatively they can be thought as lexical realizations that signal discourse relations that are not part of the closed set of `Explicit`. For example in (Ex. 3),

(Ex. 3) *It said* the delay resulted from **difficulties in resolving its accounting of a settlement with the Federal Trade Commission.**

the phrase *the delay resulted from* signals a relation of causality using an `AltLex` because the use of an `Explicit` DC such as *because* would lead to a redundant expression of the causality relation.

This thesis aim to identifying DCs with textual elements (i.e. `Explicit` and `AltLex`) in a multi-lingual context.

1.1 Goals of the Thesis

The goal of this thesis is to develop a multi-lingual approach to the identification of `Explicit` and `AltLex` DCs in the context of the Discourse Relation Parsing and Treebanking (DISRPT) 2021 shared task. The DISRPT shared task has been organized by the NLP community since 2019 to advance the state of the art in computational discourse parsing. We were particularly interested in the track of Discourse Connective Identification across Languages, which aim to identify the location of discourse connectives in texts. The most recent attempts at multilingual discourse connective identification have achieved very good results (F-measure of 92.02 for English, 94.11 for Turkish and 87.52 for Chinese) (Gessler et al., 2021), but the methods used are typically complex and require many handcrafted features. In this thesis, we will show how a simpler approach that does not rely on linguistic features can achieve similar performances, using the data sets and evaluation metrics of DISRPT 2021. We will develop and experiment with DC identification models based on transformer embeddings and different classification heads that perform well in multiple languages. Results with the DISRPT-2021 data set show that our approach is able to achieve a similar performance in multilingual discourse connective identification as the SOTA approach (Gessler et al., 2021). Our hypothesis is that BERT and BERT like transformer models already learn relevant discourse information in their pre-training tasks, and therefore fine-tuning these models simply aligns them to the task of DC identification. We show this by creating DC identification models in English, Turkish, and Chinese, which achieve strong performances even when we reduce the amount of training data available. As part of our experiments, we also created synthetic data to augment the training corpus as a means to improve the performance on low-resource languages. However, the synthetic data sets created by our methods seem to be of questionable quality and do not lead to an increase in performance. Following an inspection of these data sets, we propose ways of improving their quality.

1.2 Contributions

This thesis presents a number of theoretical and practical contributions. Note that contributions 1 to 3 below are the focus of our recently published paper at the 27th International Conference on Natural Language & Information Systems (NLDB-2022) (Muermans and Kosseim, 2022).

(1) The implementation of various DC identification models using transformers as a embedding that reach SOTA performance, yet do not need any handcrafted features (see Chapter 3).

(2) The creation of synthetic Turkish and Chinese corpora with DC annotations (see Chapter 3).

(3) Experimentation with the synthetic Turkish and Chinese corpus, in order to augment the data available for those languages and to see how they impact performance (see Chapter 4).

(4) Experimentation with corpus reduction to determine how much training data is needed to create models that still perform well (see Chapter 4).

(5) An analysis of the models' results in order to understand which DCs are well identified and which the models struggle with (see Chapter 4).

(6) An analysis of the synthetic Turkish and Chinese data sets and proposals to improve their quality (see Chapter 4).

1.3 Thesis Structure

This chapter motivated the importance of multilingual discourse connective identification and how it can help improve other NLP tasks. The rest of this thesis is structured as follows: Chapter 2 reviews the theory of the models and methods that are important to better understand the work presented. In particular, we describe the Penn Discourse Treebank framework and how it was implemented in other languages such as Chinese and Turkish, we present the data sets used in our experiments, and how we evaluated the performance of our models. Lastly, we present previous work on multilingual discourse connective identification. Chapter 3 details the models we developed for the task, as well as the procedure for creating synthetic datasets for Chinese and Turkish. Chapter 4 shows the results of our experiments and present an in-depth analysis of the results, showing where the models struggle and how well they performs on individual DCs for each language, identifying errors in the synthetic data sets and proposing ways to improve them further. Chapter 5 presents our conclusions and future work.

Chapter 2

Related Work

The goal of this thesis is to develop a multilingual discourse connective identification in the context of the Discourse Relation Parsing and Treebanking (DISRPT) [1] 2021 shared task. As such, the approach is based on data annotated with the PDTB framework, and the languages in question are English using the Penn Discourse Treebank, Chinese using the Chinese Discourse Treebank, and Turkish using the Turkish Discourse bank. In this chapter, we will briefly describe each corpora in Section 2.2, and because this work was framed within the DISRPT shared task, we will describe how the organisers use the corpora for the task in question (Section 2.3.1), and how the task is evaluated and the metrics used (Section 2.3.2). We will then describe the models (Section 2.1) and methods (Section 2.1.2) that will be of importance to this work, and finally we will discuss past work on multilingual discourse connective identification (Section 2.3.3).

2.1 Basic NLP Techniques

Neural approaches have gained popularity and have been achieving state-of-the-art performances for many natural language processing tasks over the last decade as availability to computation resources have increased and tools/frameworks have become more easily available and easy to use. Before describing specific neural approaches to DC identification, let us first describe the basic neural networks used by these approaches.

[1] for more information go to: https://sites.google.com/georgetown.edu/disrpt2021/home

2.1.1 Annotation Projection

Corpus augmentation has been shown to improve many NLP tasks where annotated data sets are scarse. In particular, annotation projection has shown its usefulness for many tasks, such as part-of-speech tagging (Yarowsky et al., 2001), word sense disambiguation (Bentivogli and Pianta, 2005), dependency parsing (Tiedemann, 2015) and discourse relations identification (Laali and Kosseim, 2017). Since they are semantic and rhetoric in nature, it is often assumed that discourse annotations can be projected from one language to the another through word alignment. In particular, Laali (2017) created a PDTB styled discourse corpus for French, by projecting discourse annotation from English (the PDTB) to French and using statistical word-alignment to identify unsupported annotations that should not be projected. The resulting corpus improved the performance of their French DC parser by 15%. Given the success of annotation projection for discourse analysis, we investigated its use to create synthetic corpora for DC annotation in Turkish and Chinese (see Section 3.3).

2.1.2 Word Alignment

Word alignment as seen in Figure 2.1, maps words from one language to another language. Word alignment is useful for many NLP tasks, such as machine translation (Brown et al., 1993), typological analysis (Östling, 2015) (Lewis and Xia, 2008), and most importantly annotation projection (Yarowsky et al., 2001) (Laali, 2017). In the 1990's to 2010's, word alignment was done using statistical word aligners such as Giza++ (Och and Ney, 2003). Then with the rise of neural networks, several attempts have been made to develop neural word alignment (Peter et al., 2017) (Garg et al., 2019), however these require large parallel training corpora which can be difficult to come by. SimAlign (Jalili Sabet et al., 2020) attempts to remedy this problem by using existing static multilingual embedding trained on non-parallel data in an unsupervised manner, such as multilingual BERT. This method has shown to match or outperform state-of-the-art word alignment results for the languages tested. As shown in Section 3.3, Word alignment will be used to project annotations from an English parallel corpus to Turkish corpus and Chinese Corpus.

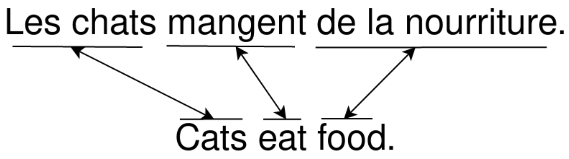


Figure 2.1: Example of word alignment French and English

2.1.3 Conditional Random Fields

Similarly to RNNs (see Section 2.1.4) and LSTMs (sec Section 2.1.4.1), Conditional Random Fields (CRF) (Lafferty et al., 2001) are appropriate for sequence labelling tasks. CRFs model a conditional probability distribution over input sequences x and label sequences y, $p(y|x)$. They are able to capture the dependencies between label predictions by building a graphical model that takes context into account. An important thing to note is that a CRF does not model the dependencies that may exist in the input x, therefore it does not model the marginal $p(x)$. The formal definition of a CRF on (x, y) is as defined by Sha and Pereira (2003):

$$p_\lambda(y|x) = \frac{exp\ \lambda \cdot F(y, x)}{Z_\lambda(x)} \tag{1}$$

where λ is the weight vector, y is the label sequence, x is the input sequence, $F(y, x)$ are feature functions (given by Equation 2), and $Z_\lambda(x)$ is a partition function (given by Equation 3):

$$F(x, y) = \sum_i f(y, x, i) \tag{2}$$

where i ranges over the input positions, and $f(y, x, i)$ is a feature function, which can be anything so long as the input that that function are y (target), x (input) and i (input position).

$$Z_\lambda(x) = \sum_y exp\ \lambda \cdot F(y, x) \tag{3}$$

To get the most likely predicted label sequence for input sequence x:

$$\hat{y} = \arg\max_y p_\lambda(y|x) = \arg\max_y \lambda \cdot F(y, x) \tag{4}$$

We can train a CRF maximizing the log-likelihood of a given training set $T = (x_k, y_k)_{k=1}^{N}$, where the maximum is attained when the empirical average of the global future vector is equal to its model expectation.

$$\mathcal{L}_\lambda = \sum_k log\ p_\lambda(y_k|x_k) = \sum_k [\lambda \cdot F(y_k, x_k) - log\ Z_\lambda(x_k)] \tag{5}$$

To optimize, we seek the zero gradient which will occur as mentioned before where the average global features vector is equal to its model expectation:

$$\nabla \mathcal{L}_\lambda = \sum_k [F(y_k, x_k) - E_{p_\lambda(y|x_k)} F(y, x_k)] \tag{6}$$

CRFs have been used for a wide variety of NLP tasks, including part-of-speech tagging, where Lafferty et al. (2001) showed that CRFs outperformed other graphical models (Hidden Markov Model, maximum entropy Markov models) for this task. Sha and Pereira (2003) showed it could be used for shallow parsing for NP chunking.

2.1.4 Recurrent Neural Networks

As text is data with sequential dependencies, it is natural to use a neural network that can learn these sequential dependencies. Recurrent neural networks (RNN) do this by allowing the layers of the network to have connections to themselves and to the layers before and after them. The formulation of a vanilla RNN is as follows:

$$h_t = f_h(W_h x_t + U_h h_{t-1} + b_h) \tag{7}$$

$$y_t = f_y(W_y h_t + b_y) \tag{8}$$

where x_t is the input vector (eg. a word in a sentence), h_t is the hidden state, y_t is the output vector (eg. a label for a word), W, U are trainable parameter matrices, and b (the bias) is a trainable vector. f_h and f_y are non-linear activation functions such as ReLU, softmax, sigmoid.

Although RNNs have many benefits compared to feedforward neural networks for sequential data, they do have drawbacks. RNNs cannot be parallelized during training, as one input at

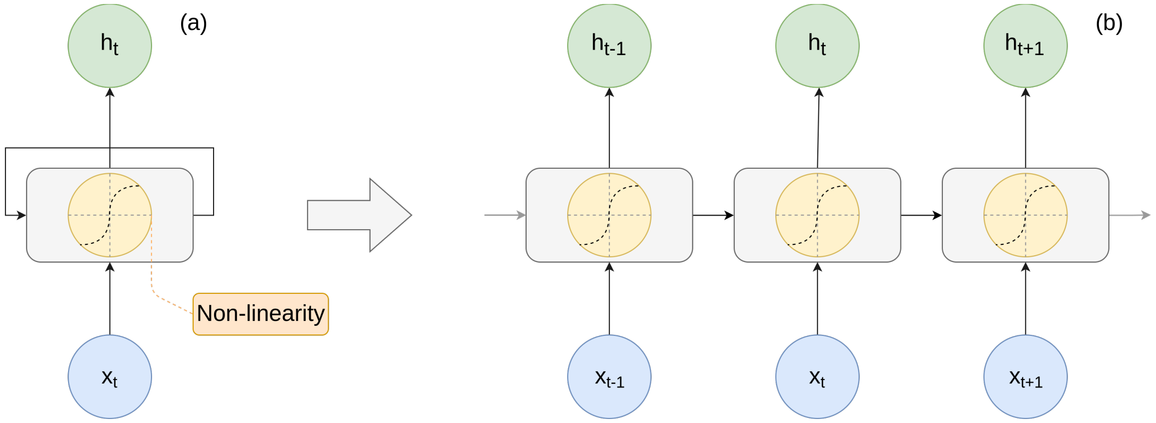


Figure 2.2: Recurrent neural network: (a) Recurrent network presented with the self-loop (b) Unrolled presentation of RNN with respect to time. (Davari, 2020)

time `t` needs the result of time `t-1`, which causes the training procedure to be time consuming. Additionally, vanilla RNNs have difficulty with long input sequences. These can cause a vanishing or exploding gradient problem as shown by Bengio et al. (March 1994); Hochreiter and Schmidhuber (November 1997), and effectively stops the RNN from learning anything. To address this issue Hochreiter and Schmidhuber (November 1997) proposed the Long Short-Term Memory (LSTM) architecture.

2.1.4.1 Long Short-Term Memory

In order to address the vanishing and exploding gradient problems, Hochreiter and Schmidhuber (November 1997) proposed a modification to the vanilla RNN unit, Long Short-Term Memory (LSTM), that allows the network to remember over an arbitrary sequence length. This process is done using an input gate, a forget gate, and an output gate. Figure 2.3 shows the architecture of an LSTM unit. As the bottom portion of the figure shows, the previous output h_{t-1} is combined with the current input x_t; using a Sigmoid (σ) this is similar to the vanilla RNN using a Sigmoid activation. The top portion addresses the vanishing and exploding gradient problems by allowing the network to select what to remember from the previous state and the current state. The formal definition of an LSTM unit is shown in Equations 10, 10, and 11, where i is the value of the input gate, f is the value of the forget gate, and o is the value of the output gate. All these values are

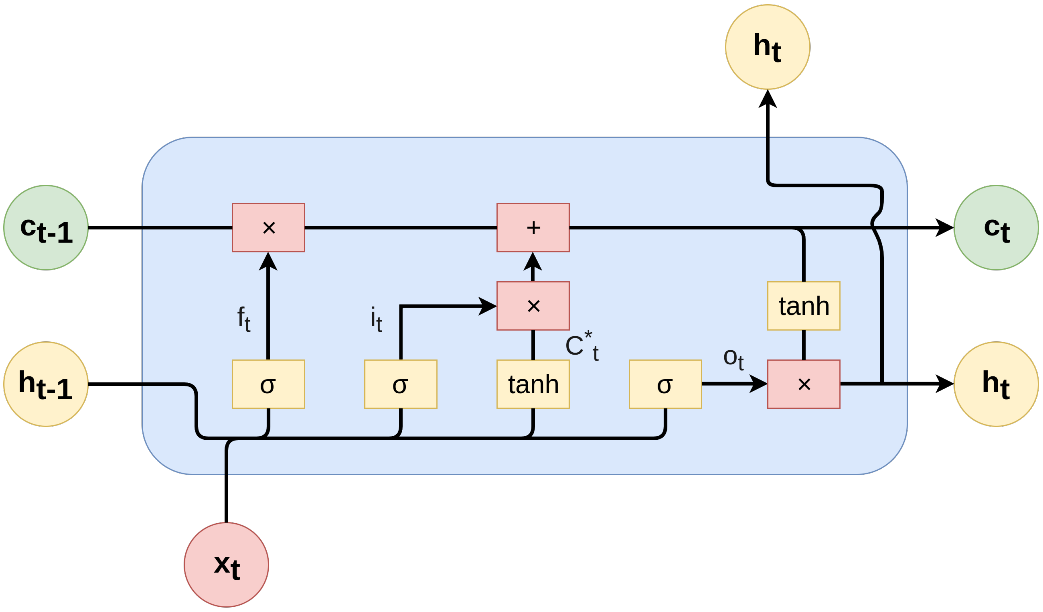


Figure 2.3: LSTM architecture (Davari, 2020)

computed at time t.

$$i_t = \sigma(U^i x_t + W^i h_{t-1}) \tag{9}$$

$$f_t = \sigma(U^f x_t + W^f h_{t-1}) \tag{10}$$

$$o_t = \sigma(U^o x_t + W^o h_{t-1}) \tag{11}$$

where x_t is the input at time iteration t, h_{t-1} is the output of the LSTM unit at the previous time step, Us are the matrices of weights connecting the input to the LSTM unit, Ws are the matrices of weights used for the internal connections of the LSTM, and σ is the Sigmoid activation function. The final output of the LSTM at time t, h_t, is computed as the dot product ($\odot$) between the cell state (what to remember) and the output of the Sigmoid:

$$h_t = \tanh(C_t) \odot o_t \tag{12}$$

where C_t is called the *cell state* and is computed via the following 2 equations:

$$C_t^* = \tanh(U^g x_t + W^g h_{t-1}) \tag{13}$$

$$C_t = \sigma(f_t \odot C_{t-1} + i_t \odot C_t^*) \tag{14}$$

While the LSTM does result in better performances over long sequences compared to the vanilla RNN, it introduces a new problem, that is all these new weight matrices and additional computation make it expensive to train in time and in computational resources. In addition, the LSTM does not solve the lack of parallelization issue of the vanilla RNN. To address this issue a new architecture must be used, the Transformer.

2.1.5 Transformers

Transformers address some of the issues of RNNs. As seen in Section 2.1.4, RNNs are not parallelized; meaning the training is prohibitive yet computational resources are left unused. Transformers (Vaswani et al., January 2017) aim to remedy this in their architecture. The Transformers architecture is based only on the attention mechanism. Attention allows a network to focus on what is important and the dependencies between inputs and target outputs. The algorithm achieves this by distributing weights to the components of an input; for an NLP task this would in the input words, which allows it to model long term dependencies. Over a sequence of length n, an RNN has to essentially loop through the entire sequence. This leads to $O(n)$ computation complexity over the sequence, whereas Transformers are parallelized over the sequence, meaning they have a $O(1)$ complexity. The complexity of the transformer also determines the length of the longest path that the model has for modeling long term dependencies and as Bengio et al. (March 1994); Bahdanau et al. (May 2015) showed, longer paths prevent gradient error signals from being propagated. Transformers having a longest path of $O(1)$ easily handle long term dependencies. Based on Vaswani et al. (January 2017) attention in Transformers is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{n}}\right)V \tag{15}$$

where K and V are key-value pairs that represent the encoding; and Q, the query, is the output of the decoder. Transformers perform this attention operation multiple times, leading to the name Multi-head attention, which allows the model to attend to different dependencies within the sequence. For a natural language sentence, these dependencies might be semantic (eg. discourse relations), or syntactic (eg. part-of-speech) dependencies. The number of times the attention operation is performed depends on the number of heads h:

$$\mathrm{MultiHead}(Q, K, V) = [\mathrm{head}_1, \ldots \mathrm{head}_h]W^O \tag{16}$$

$$\text{where } \mathrm{head}_i = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{17}$$

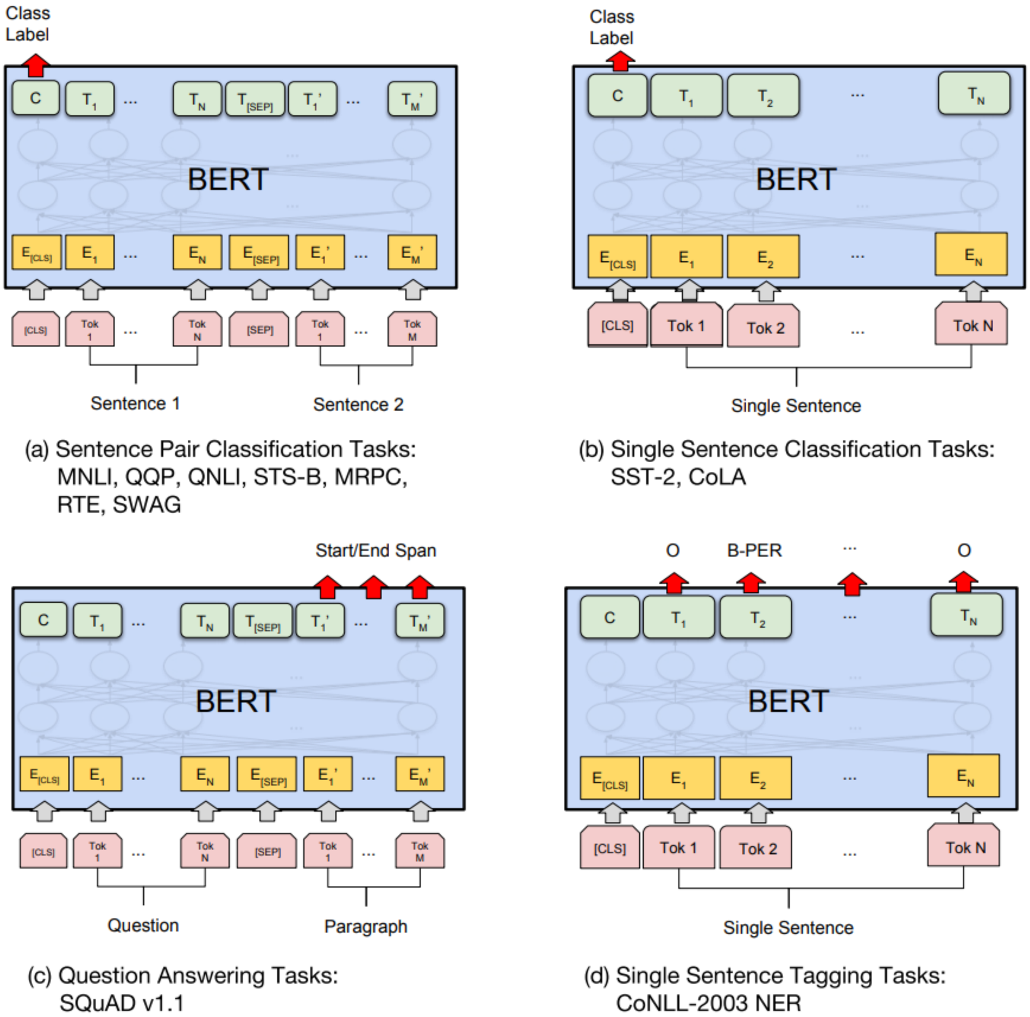
where W^O, W_i^Q, W_i^K, and W_i^V are weight matrices that will be adjusted as training is done.

2.1.5.1 Bidirectional Encoder Representations from Transformers

Devlin et al. (2018) proposed the Bidirectional Encoder Representations from Transformers (BERT) model, which is a pre-trained language model using a multi-layer bidirectional transformer encoder architecture. BERT models help to alleviate the problem of designing task specific models, by proposing a model that is pre-trained in an unsupervised fashion, then using this model as an embedding and fine-tuning it for a specific task. BERT is pre-trained on two language modeling tasks: masking language model (MLM), designed to understand the context around a word, and next sentence prediction (NSP), which seeks to model the relation between two sentences. As Figure 2.4 shows, once the model is trained on a large corpus, it can be fine-tuned with ease by changing what is inputted into the model and which output is important for the downstream task. Sections 3.2.1, 3.2.2, 3.2.3, 3.2.4, will describe how we use BERT in our work. Given the success of BERT in many NLP tasks, several more specific models have been developed as follow-ups.

2.1.5.2 Generative Pre-trained Transformer 2

Radford and Narasimhan (2018) proposed a Generative Pre-trained Transformer (GPT) model whose goal is to learn a universal representation that transfers to a wide range of tasks with little adaptation. The model is pre-trained in two stages. The first stage is a language modeling objective



Class Label
C
T1
...
TN
T[SEP]
T1'
TM'
BERT
E[CLS]
E1
EN
E[SEP]
E1'
EM'
[CLS]
Tok 1
Tok N
[SEP]
Tok M
Sentence 1
Sentence 2
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG
Class Label
T2
E2
Tok 2
Single Sentence
(b) Single Sentence Classification Tasks: SST-2, CoLA
Start/End Span
Question
Paragraph
(c) Question Answering Tasks: SQuAD v1.1
O
B-PER
O
Single Sentence
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

Figure 2.4: Fine-tuning BERT in downstream tasks with few new training parameters added to the base model and slightly modified training objective. (Devlin et al., 2018)

on unlabeled data, which initialized the model parameters. In the second stage, the model is adapted to a target task using a corresponding supervised objective. GPT-2 (Radford et al., 2019) has a similar objective, base transformer architecture and training methods as GPT. However, it has an order of magnitude more learning parameters than GPT while also being trained on more data. GPT-2 should be able to generalized very well to a wide variety of tasks given its size. Section 3.2.6 will describe how we used GPT2 in our work.

2.1.5.3 A Robustly Optimized BERT Pretraining Approach

Liu et al. (2019) introduced a Robustly Optimized BERT Per-training Approach (RoBERTa), because they found that BERT was under-trained in its per-training tasks. The resulting RoBERTa model achieved at the time state-of-the-art on GLUE[2], RACE[3], and SQuAD[4]. To achieve these results they trained the model longer, with larger batches, over more data, in addition, they removed the next sentence prediction objective, trained on longer sequences and dynamically changed the masking pattern applied to the training data. While the architecture of RoBERTa remains the same as BERT, it is better optimized. Therefore using this model should result in better performances over using BERT. Section 3.2.5 will describe how we used RoBERTa in our work.

2.1.6 Tokenization

Tokenization is an important step in NLP tasks, and can be done in many different ways. Traditionally, tokens were made from words and punctuation separated by white spaces or other punctuation marks. This worked well enough for languages similar to English, but was insufficient for languages where punctuation is different and spacing between words may not exist (eg. Chinese, Japanese). With these languages, this approach results in large vocabularies that are incapable of handling out-of-vocabulary words. To fix these issues, new tokenization methods that rely on sub-words have been proposed. These approaches breakdown complex or rare words into multiple sub-units that still retain some of their meaning, resulting in a smaller and more flexible vocabulary. There are many ways of doing sub-word tokenization, the two we are interested in are:

[2] https://gluebenchmark.com/

[3] https://www.cs.cmu.edu/~glai1/data/race/

[4] https://rajpurkar.github.io/SQuAD-explorer/

- WordPiece algorithm (Wu et al., September 2016).
- Byte Pair Encoding (BPE) algorithm (Sennrich et al., August 2016).

BERT uses the WordPiece algorithm, which first creates a vocabulary with every character in the training set and progressively learns a given number of merge rules that maximized the likelihood of the training set, if a “character symbol pair probability divided by the probabilities of its first symbol followed by its second symbol is the greatest among all symbol pairs.”[5]

Whereas, RoBERTa and GPT-2 use BPE, which uses a set of unique words and their frequencies to build a vocabulary of symbols and learns merge rules to form new symbols from pairs of symbols in the base vocabulary, then counts which symbol pair are most frequent, and this process is repeated until the vocabulary reaches a desired size.

2.2 Penn Discourse Treebank Framework and Corpora

The Penn Discourse Treebank (PDTB) framework (Prasad et al., 2008) is one of the most widely used annotation scheme developed to facilitate research in computational discourse analysis by providing strict annotations guidelines to create annotated texts labeled with discourse connectives, discourse arguments, and discourse relations. The PDTB framework was originally used to create an annotated corpus for English, called the PDTB corpus (see Section 2.2.1), which was annotated by linguists and reached a very high annotator agreement; then several corpora following the PDTB framework were created, including the CDTB for Chinese (see Section 2.2.2) and the TBD for Turkish (see Section 2.2.3).

2.2.1 English PDTB Corpus

The Penn Discourse Treebank (PDTB) (Prasad et al., 2008) corpus was created using the Penn Treebank (PTB) (Marcus et al., 1993) corpus, itself composed of articles from the Wall Street Journal (WSJ). The PDTB is based on the simple idea that discourse relations are signaled by a set of words or phrases called discourse connectives (DC) or in adjacent sentences for `Implicit` relations. The PDTB annotates the beginning and the end of two textual units related by a discourse

[5] https://huggingface.co/docs/transformers/tokenizer_summary

relation. These textual units are known as arguments, Arg1 and Arg2, which may include a DC. The discourse relation between the two arguments is labeled with a *sense*. Figure 2.5 shows a hierarchy of the possible *senses* in the PDTB 3.0.

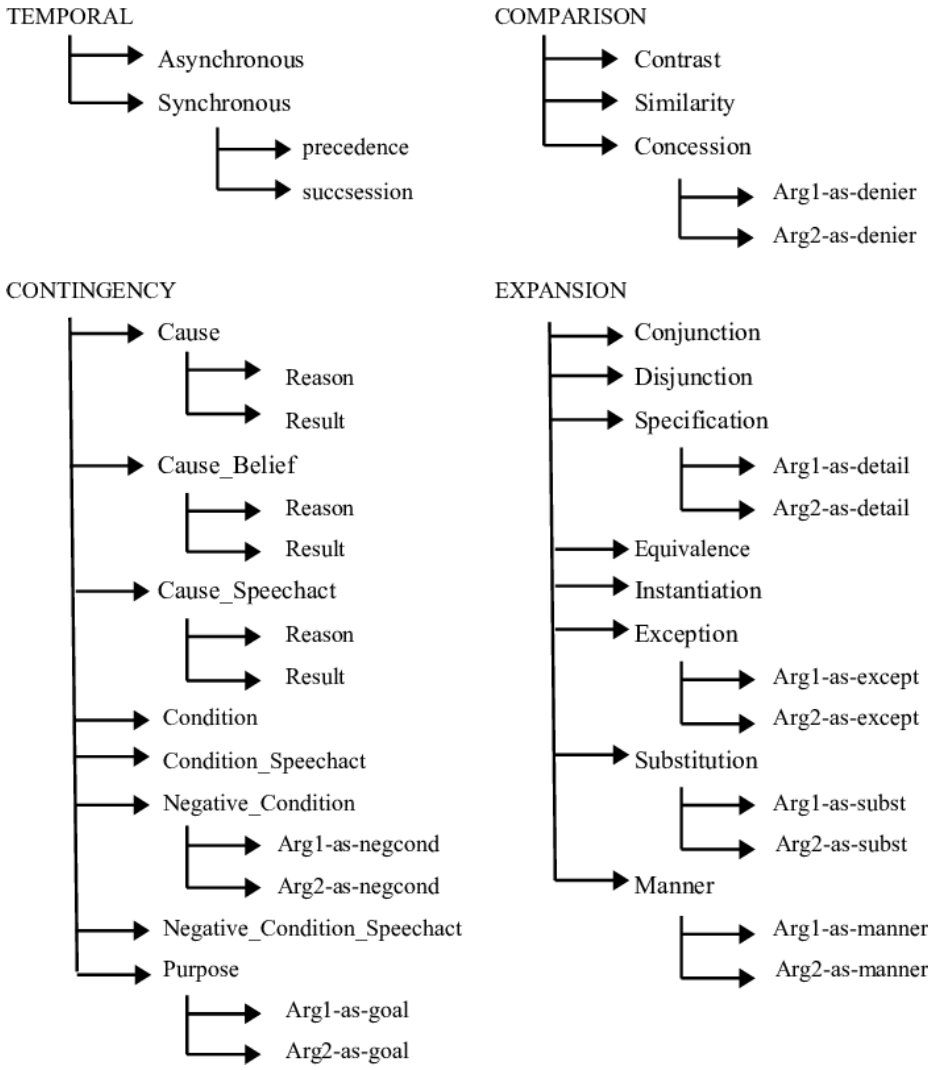


Figure 2.5: PDTB 3.0 sense hierarchy (Rehbein et al., 2016)

Discourse relations in the PDTB can be signaled by:

(1) Explicit relations: (Ex.) *The city's Campaign Finance Board has refused to pay Mr. Dinkins $95,142 in matching funds* because **his campaign records are incomplete.**

(2) Implicit relations: (Ex.) *The city's Campaign Finance Board has refused to pay Mr. Dinkins $95,142 in matching funds.* **His campaign records are incomplete.**

	Training	**Validation**	**Test**
Explicit	14722	680	923
Implicit	13156	522	769
AltLex	524	19	30
EntRel	4133	215	217
NoRel	204	8	4
Total	**32739**	**1444**	**1943**

Table 2.1: Statistics of the PDTB 3.0.

(3) Alternative lexicalizations `AltLex`: (Ex.) *It said* the delay resulted from **difficulties in resolving its accounting of a settlement with the Federal Trade Commission.**

(4) Entity relations `EntRel`: (Ex.) *Hale Milgrim, 41 years old, senior vice president, marketing at Elecktra Entertainment Inc., was named president of Capitol Records Inc., a unit of this entertainment concern.* **Mr. Milgrim succeeds David Berman, who resigned last month.**

(5) No Relation `NoRel`: (Ex.) *Mr. Rapanelli met in August with U.S. Assistant Treasury Secretary David Mulford.* **Argentine negotiator Carlos Carballo was in Washington and New York this week to meet with banks.**

`Explicit` relations are signaled by a well defined closed set of 100 DCs (e.g. but, if, because...), these 100 DCs more frequently signal discourse relations over other terms. Whereas `Implicit` relations are not signaled by a textual element, but rather they are inferred by the context. Similarly to `Implicit`, `AltLex` are not signaled by a closed set of DCs but by a more flexible textual expression, any discourse relation that is signaled by textual realization that is not part of the 100 DCs is an `AltLex` DC. `EntRels` are signaled via an entity-based coherence relation and are closely related to `Implicit` relations. Lastly `NoRel`, where an implicit connective could not be provided.

Table 2.1 shows statistics of the PDTB 3.0. The PDTB is split into 24 sections. The PDTB manual suggests using sections 02 to 21 for training, sections 0, 1 and 24 for validation, and section 23 for testing. This section splitting is often called the PDTB-split.

In the PDTB, `Explicit` and `Implicit` relations are annotated separately. For `Explicit` relations, the first step involves finding the DC. DCs can be characterized by 3 syntactic functions:

(1) Subordinating conjunctions: (Ex.) *The cyclist took the traffic heavy boulevard* because **they did not have safer alternative routes.**

(2) Coordinating conjunctions: (Ex.) *They lived in Montreal,* and **went to study at Concordia University**.

(3) Discourse adverbials: (Ex.) Anyway **, I need to go and pick up lunch.**

It is important to note that some DCs may be used in a discourse usage (DU) (see Ex. (Ex. 4)) or in a non-discourse usage (NDU) (see (Ex. 5)). This is a source of ambiguity in the annotation procedure. In (Ex. 4) 'and' is in discourse usage (DU) because it links the two arguments by an expansion-conjunction relation; whereas, in (Ex. 5) 'and' is used in an itemized list (i.e. NDU).

(Ex. 4) *It employs 2,700 people* and **has annual revenue of about $ 370 million.**

(Ex. 5) My favorite languages are Python and Go.

Once the DC is located, its arguments `Arg1` and `Arg2` need to be identified, this is highly dependent on the physical location to the connective.

Annotating `Implicit` relations follows the same principle as `Explicit` relations, except that the DC is not present. The goal of the annotators was, therefore, to identify where a DC could have been located. Because this task is not always easy, constraints were imposed. `Implicit` relations can only occur within a sentence bounded by a period, a question mark, an exclamation mark, or a semicolon or within a pair of sentences which are adjacent and do not cross paragraph boundaries. Once a relation is identified, the annotator inserts a DC that could convey the same discourse relation. If no DC is applicable, they must identify if the relation is signaled by `AltLex`, `EntRel`, or `NoRel`.

2.2.2 Chinese Discourse Treebank (CDTB)

The Chinese Discourse Treebank (CDTB) (Zhou and Xue, 2015) is a subset (70k words) of the Chinese Treebank (4.1 million words), where discourse relations have been annotated following the PDTB framework with some alterations. The most important changes include the adoption of a flat *sense* inventory containing 11 classes (Alteration, Causation, Conditional, Conjunction, Contrast, Expansion, Purpose, Temporal, Progression, EntRel, and NoRel). This was done to facilitate the task of the annotators given the specificities of the Chinese language itself. Unlike the PDTB, where `Explicit` and `Implicit` relations were annotated separately, these relations were marked simultaneously in the CDTB. This was done for a variety of reasons, but the main one being that in Chinese `Explicit` discourse relations only account for 22% of all discourse relations, as it is very common to drop the DCs in Chinese; whereas in English, `Explicit` relation account for 45.47% of the PDTB. Table 2.2 shows statistics of the CDTB.

Identifying `Explicit` relations in Chinese is similar to that of English: First, a DC must be found, characterized by the same 3 syntactic functions, but Chinese has an additional function called “localizers”. Subordinating conjunctions, coordinating conjunctions, and “localizers” have two lexical realizations: single and paired. Paired DCs, also known as discontinuous DCs, consist of single DCs fragmented with a textual span between the segments (e.g. if ... then). It is interesting to note that while English has a very small number of discontinuous DCs, most DCs in Chinese are discontinuous, though part of the pair can be dropped and would still be considered a DC. In two Chinese common DC lexicons, (王起澜et al., 1989) contains 1165 DCs where 62.7% are discontinuous DCs and (戴木金et al., 1988) contains 1344 DCs where 77.3% are discontinuous DC. Unlike the PDTB, the identification of Chinese `Arg1` and `Arg2` is not based on their physical location but rather on semantics, which is different for each *sense*. This makes the annotation consistent between all forms of DCs: `Explicit` relations (single, paired, or if only part of a pair is present) and `Implicit` relations.

Annotating `Implicit` relations in Chinese is different than in English. This is mainly because of the way Chinese sentence boundaries are conventionalized. Unlike English, commas can often indicate a sentence boundary in Chinese. This means that if `Implicit` relations are to be properly identified, intra-sentential `Implicit` relations separated by a comma need to be annotated. Similarly to English, the annotators were asked to insert a DC that best fits the relation being signaled. If none were suitable, the annotators identified it as either an `AltLex`, `EntRel`, or `NoRel`.

	Count	**Ratio**
Explicit	1223	22.10%
Implicit	4193	75.80%
AltLex	118	2.10%
EntRel	0	0.00%
NoRel	0	0.00%
Total	**5534**	**100%**

Table 2.2: Statistics of the CDTB.

2.2.3 Turkish Discourse Bank (TDB)

The Turkish Discourse Bank (TDB) (Zeyrek et al., 2013) (Demirşahin and Zeyrek Bozşahin, 2017) is built on a subset of the METU Turkish Corpus (MTC) (Say et al., 2002). The TDB contains texts from various genres (novels, stories, research articles, essays, travel, interviews, diaries, memoirs, news) written from 1990 to 2000. The TDB is also based on the PDTB framework, where two text spans `Arg1` and `Arg2` are arguments to a DC. Similarly to English, DCs belong to three syntactic classes:

(1) Conjunctions (coordinating and other). (`Explicit`)

(2) Subordinators/Subordinating Conjunctions (complex). (Ex.) *için* (for), *karşın* (although/despite') (`Explicit`)

(3) Discourse adverbials. (`Explicit`)

(4) Phrasal expressions. (`AltLex`)

The TDB does not annotate simplex subordinators, i.e converbs (*-IncA* 'when,' *–ken* 'while, now that'). In the TDB 1.0, only `Explicit` and phrasal expressions (a form of `AltLex`) are annotated with their two arguments; their sense and other discourse relations were left for future work. The TDB 1.1 takes 10% of the TDB 1.0 and adds `AltLex`, `EntRel`, `Implicit` where the best fit DC is inserted, as well as *sense* to `Explicit`, `AltLex`, and `Implicit`. For the task of DC identification we are mainly interested in `Explicit`, and `AltLex`, and because we do not need the *sense*, we safely used the TDB 1.0. The Table 2.3 shows statistics of the TDB 1.0.

Syntactic Class	**Count**	**Ratio**
Explicit	7789	94.18 %
Implicit	0	0.00%
AltLex	494	5.82%
EntRel	0	0.00%
NoRel	0	0.00%
Total	**8483**	**100%**

Table 2.3: Statistics of the TDB.

2.2.4 Other PDTB-style Corpora

2.2.4.1 TED-Multilingual Discourse Bank

The TED-Multilingual Discourse Bank (TED-MDB) (Zeyrek et al., 2019, 2018), is a parallel corpus of TED talks transcriptions with PDTB styled DC annotations, identifying `Explicit`, `Implicit`, `AltLex`, `EntRels`, and `NoRels`. The corpus contains 7 languages: English, German, Lithuanian, Polish, Portuguese, Russian and Turkish. The annotation was done manually and in a similar way to the original PDTB corpus (see Section 2.2), though several types of DCs that tend to appear in speeches, such as attribution, pragmatic markers and modified connectives are not annotated. This dataset is fairly small containing only 424 sentences and is not in the same format as the DISRPT 2021 dataset, making it difficult to use for our research.

2.3 DISRPT 2021 Shared Task

2.3.1 DISRPT 2021 Datasets

The DISRPT 2021 (Zeldes and Liu, 2021) shared task provides a script to convert/move the PDTB, TDB and CDTB raw textual data into three different formats: *conllu*, *tok*, and *rels*. For the task of DC identification, we are only interested in `Explicit` and `AltLex` as these are the only relations that are signaled via a textual element. The *conllu* format[6] provides plain text annotations of various linguistic features. Listing 2.1 shows a sample from the *conllu* training set. As shown in Listing 2.1, each word is annotated with the word lemma, universal part-of-speech, language-specific part of speech, list of morphological features, head of the current word, universal

[6] see `https://universaldependencies.org/format.html` for a breakdown of each field

dependency relations, enhanced dependency graph and a section for any other annotation. The DISRPT 2021 shared task uses the section for any other annotation to signal the presence of a DC or the lack thereof. A DC is annotated with the label `Seg=B-Conn` which signals that the word is the beginning of a DC, and `Seg=I-Conn` which signals that the word is inside of a DC. The example in Listing 2.1 shows the `Explicit` DC 'in fact' annotated with `Seg=B-Conn` and `Seg=I-Conn`. The *conllu* files provided by the DISRPT 2021 shared task are split by sentence and the words/punctuation are tokenized, this makes the task of DC identification into one of token classification, where the model needs to predict a tag for each token. Table 2.4 shows statistics of each *conllu* file. The *tok* file similarly tokenizes the words/punctuation but does not split the sentences in the raw text, nor does it provide any additional linguistic features. It annotates DCs in a similar way to the *conllu* files, signaling that the word is the beginning of a DC with `Seg=B-Conn` and a word is inside a DC with `Seg=I-Conn`. The *rels* file provides annotations on the *sense* of the relation being signaled by the DC.

Corpus	Language	# of Sentences	# tok `B-Conn` + `I-Conn`	% tok `B-Conn` + `I-Conn`
PDTB-train	English	44,563	28,349	2.671
PDTB-dev	English	1703	1,112	2.796
PDTB-test	English	2364	1,483	2.665
TDB-train	Turkish	24,960	7,572	1.900
TDB-dev	Turkish	2,948	888	1.777
TDB-test	Turkish	3,289	919	1.919
CDTB-train	Chinese	2,049	1,171	2.249
CDTB-dev	Chinese	438	398	3.560
CDTB-test	Chinese	404	354	3.514

Table 2.4: Statistics of the *conllu* training, validation and test data at DISRPT 2021.

For the purpose of multilingual DC identification we used the *conllu* files.

Listing 2.1: Example of a PDTB *conllu* file (eng.pdtb.pdtb_train-1677)

```
# sent_id = eng.pdtb.pdtb_train-1677
# s_type = decl
# text = In fact , " the market has always tanked .
1 In  in  ADP IN  _ 2 case  2:case  Seg=B-Conn
2 fact  fact  NOUN  NN  Number=Sing 9 obl 9:obl:in  Seg=I-Conn
3 , , PUNCT , _ 9 punct 9:punct _
4 " ``  PUNCT ``  _ 9 punct 9:punct _
5 the the DET DT  Definite=Def|PronType=Art 6 det 6:det _
6 market  market  NOUN  NN  Number=Sing 9 nsubj 9:nsubj _
7 has have  AUX VBZ Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin 9 aux 9:aux _
8 always  always  ADV RB  _ 9 advmod  9:advmod  _
9 tanked  tank  VERB  VBN Tense=Past|VerbForm=Part  0 root  0:root  _
10  . . PUNCT . _ 9 punct 9:punct _
```

2.3.2 DISRPT 2021 Metrics

The evaluation of DC identification models, is done using the standard metrics of precision, recall and F-measure.

Precision P measures the ratio of true positives T_p to the sum of T_p and false positives F_p and can be seen as measuring the quality of the predictions:

$$P = \frac{T_p}{T_p + F_p} \tag{18}$$

Recall R measures the ratio of T_p to the sum of T_p and false negatives F_n and can be seen as measuring the quantity of predictions:

$$R = \frac{T_p}{T_p + F_n} \tag{19}$$

Using two different metrics can make it difficult to know which predictions are better; this is why P and R are combined into a single measure; the F-measure. For the purpose of the this thesis we will only use $F1$ which is the harmonic mean of P and R:

$$\mathrm{F1} = 2 \cdot \frac{1}{\frac{1}{\mathrm{recall}} + \frac{1}{\mathrm{precision}}} = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{20}$$

The evaluation script provided by DISRPT 2021 uses a strict exact match of the DC annotation and no points are given for partial identification. To better understand how the evaluation script works, Listing 2.2 shows an example gold annotation for the generic sentence “A B C D E F.” Listings 2.3 to 2.8 show possible predictions for this sentence. We will look at the P, R and $F1$ for each.

Listing 2.2: Sample Gold Annotation

```
# text = A B C D E F .
1 A _ _ _ _ _ _ _ _
2 B _ _ _ _ _ _ _ Seg=B-Conn
3 C _ _ _ _ _ _ _ Seg=I-Conn
4 D _ _ _ _ _ _ _ _
5 E _ _ _ _ _ _ _ Seg=B-Conn
6 F _ _ _ _ _ _ _ _
7 . _ _ _ _ _ _ _ _
```

Listing 2.3: Sample Prediction 01: No DC identified

```
# text = A B C D E F .
1 A _ _ _ _ _ _ _ _
2 B _ _ _ _ _ _ _ _
3 C _ _ _ _ _ _ _ _
4 D _ _ _ _ _ _ _ _
5 E _ _ _ _ _ _ _ _
6 F _ _ _ _ _ _ _ _
7 . _ _ _ _ _ _ _ _
```

The sample prediction 01, shown in Listing 2.3 annotates none of the words as DCs. Hence its performance can be calculated as $P = \frac{0}{0+0} = 0.0\%$, $R = \frac{0}{0+2} = 0.0\%$ and $F1 = 2 \cdot \frac{0 \cdot 0}{0+0} = 0.0\%$, as the official evaluation script is only concerned with DCs.

Listing 2.4: Prediction 02: Only a partial DC identified

```
# text = A B C D E F .
1 A _ _ _ _ _ _ _ _
2 B _ _ _ _ _ _ _ Seg=B-Conn
3 C _ _ _ _ _ _ _ _
4 D _ _ _ _ _ _ _ _
5 E _ _ _ _ _ _ _ _
6 F _ _ _ _ _ _ _ _
7 . _ _ _ _ _ _ _ _
```

Prediction 02 in Listing 2.4 identifies only one DC starting and ending at word B. Its performance will therefore be calculated as $P = \frac{0}{0+1} = 0.0\%$, $R = \frac{0}{0+2} = 0.0\%$ and $F1 = 2 \cdot \frac{0 \cdot 0}{0+0} = 0.0\%$, as only complete matches to a DC in the gold are given points.

Listing 2.5: Prediction 03: One DC identified (multi-word DC)

```
# text = A B C D E F .
1 A _ _ _ _ _ _ _ _
2 B _ _ _ _ _ _ _ Seg=B-Conn
3 C _ _ _ _ _ _ _ Seg=I-Conn
4 D _ _ _ _ _ _ _ _
5 E _ _ _ _ _ _ _ _
6 F _ _ _ _ _ _ _ _
7 . _ _ _ _ _ _ _ _
```

Prediction 03 in Listing 2.5 identifies one DC starting on B and ending on C, matching the gold annotation for that DC; but the DC at word E is not identified. Its performance therefore will be calculated as $P = \frac{1}{1+0} = 100.0\%$, $R = \frac{1}{1+1} = 50.0\%$ and $F1 = 2 \cdot \frac{1.0 \cdot 0.5}{1.0+0.5} = 66.7\%$.

Listing 2.6: Prediction 04: Extra word in DC is identified

```
# text = A B C D E F .
1 A _ _ _ _ _ _ _ _
2 B _ _ _ _ _ _ _ Seg=B-Conn
3 C _ _ _ _ _ _ _ Seg=I-Conn
4 D _ _ _ _ _ _ _ Seg=I-Conn
5 E _ _ _ _ _ _ _ _
6 F _ _ _ _ _ _ _ _
7 . _ _ _ _ _ _ _ _
```

Prediction 04 in Listing 2.6 identifies one DC starting on B and ending on D, which does not match the gold standard, D should not be identified; additionally the DC at word E is not identified. Its performance therefore will be calculated as $P = \frac{0}{0+1} = 0.0\%$, $R = \frac{0}{0+2} = 0.0\%$ and $F1 = 2 \cdot \frac{0.0 \cdot 0.0}{0.0+0.0} = 0.0\%$.

Listing 2.7: Prediction 05: One DC identified (single-word DC)

```
# text = A B C D E F .
1 A _ _ _ _ _ _ _ _
2 B _ _ _ _ _ _ _ _
3 C _ _ _ _ _ _ _ _
4 D _ _ _ _ _ _ _ _
5 E _ _ _ _ _ _ _ Seg=B-Conn
6 F _ _ _ _ _ _ _ _
7 . _ _ _ _ _ _ _ _
```

Prediction 05 in Listing 2.7 annotates a DC starting and ending at E, matching the gold annotation for that DC; but misses the DC at BC. Its performance therefore will be calculated as $P = \frac{1}{1+0} = 100.0\%$, $R = \frac{1}{1+1} = 50.0\%$ and $F1 = 2 \cdot \frac{1.0 \cdot 0.5}{1.0+0.5} = 66.7\%$.

Listing 2.8: Prediction 06: One correct and one partial DC identified

```
# text = A B C D E F .
1 A _ _ _ _ _ _ _ _
2 B _ _ _ _ _ _ _ Seg=B-Conn
3 C _ _ _ _ _ _ _ _
4 D _ _ _ _ _ _ _ _
5 E _ _ _ _ _ _ _ Seg=B-Conn
6 F _ _ _ _ _ _ _ _
7 . _ _ _ _ _ _ _ _
```

Prediction 06 in Listing 2.8 annotates the start and end of a DC at B (instead of BC) and another starting and ending at E. Its performance therefore will be calculated as $P = \frac{1}{1+1} = 50.0\%$, $R = \frac{1}{1+1} = 50.0\%$ and $F1 = 2 \cdot \frac{0.5 \cdot 0.5}{0.5+0.5} = 50.0\%$ as the partial identification does not count.

2.3.3 Previous Approaches to DC Identification

The earliest attempt at identifying DCs automatically using the PDTB dates back to Pitler and Nenkova (2009) who used extracted features from gold-standard Penn Treebank parses, and a maximum entropy classifier and obtained an F-measure of 94.19 for `Explicit` DC disambiguation on the PDTB test set. Johannsen and Søgaard (2013) showed that a simple logistic regression model could achieve better results without relying on gold-standard parse trees, using lexical features and part-of-speech tags only. Laali et al. (2016) created a complete discourse parser that first identifies `Explicit` DCs, annotated the *sense*, and segmented `Arg1` and `Arg2` automatically. To do this, they used a decision tree binary classifier to disambiguate if the connective is in discourse usage (DU) or not (NDU), then used a different decision tree to identify the *sense*, and used a conditional random field (CRF) to segment `Arg1` and `Arg2`. This work achieved 91.00 F-measure for `Explicit` DC identification, 89.48 F-measure for sense labeling of those `Explicit` DCs and 40.23 F-measure for `Explicit` argument identification on the PDTB test set. `Explicit` DC identification can be done using non-neural approaches to near human level performance, for English using the PDTB as a training set.

In 2019, Muller et al. (2019) was the team with the best performance. They employed multilingual BERT and bi-directional LSTMs and achieved F-measures of 88.60%, 69.85%, 79.32% in English, Turkish and Chinese respectively. These results seem to correlate with the size of the training set of 44k, 24k and 2k respectively. This motivated creating synthetic data to increase the performance for the lower-resource languages. The most recent attempt for the detection of multilingual DC identification, (Gessler et al., 2021) used transformer models (see Section 2.1.5) in addition to many handcrafted input features and a conditional random field (see Section 2.1.3) as a final classifier instead of a linear output layer, achieving the best performance at DISRPT 2021 with an F-measure of 92.02%, 94.11%, 87.52% for English, Turkish and Chinese respectively; leading to a significant improvement in all three languages.

In this chapter we have reviewed the PDTB framework and three datasets in different languages that employ the framework to create corpora with discourse annotation: we explained the different type of discourse annotation which can be found in these corpora `Explicit`, `Implicit`, `AltLex`, `EntRel`. We explained how the DIRSPT shared task uses this data to provide the participants with an simple and intuitive datasets to train their models on which only contains the annotation for DC that have textual elements (`Explicit`, `AltLex`), and how the evaluation script they provided works. We then explained various models such as transformers and CRF and methods (word alignment and annotation projection) that are important to the work presented in this thesis. Lastly we went over recent work in multilingual discourse connective identification, which helped guide the models we have used in this work. Chapter 3 will describe our approach in detail.

Chapter 3

Models and Methods

This chapter will explain the frameworks used to create our model architectures (see Sections 3.1); then will go into detail about the specific models employed for DC identification (see Section 3.2). An overview of the methods we developed for data augmentation: annotation projection, annotation agreement, and the union of the two will be described in Section 3.3. Followed by a description of the corpora used for the augmentation of Chinese and Turkish corpora and how the methods are implemented for the two languages. Finally the method used for data reduction will be presented in Section 3.4.

3.1 Tools and Frameworks

We have developed and experimented with 6 different models for multilingual DC identification. The details of each model will be described in Section 3.2. However, before describing the models, let us present the built-in tools and frameworks that we used to build them.

3.1.1 Huggingface

All models rely on pre-trained embeddings (see Section 2.1.5). For this, we used Huggingface[1], a community driven database of pre-trained models that are to be fine-tuned on downstream tasks. The specific embeddings used in this thesis are shown in Table 3.1. As the table shows,

[1] https://huggingface.co/

most embeddings are monolingual, while one model is multilingual. The multilingual model, bert-base-multilingual-cased, is pre-trained on 104 languages, and has been used in our research to validate if multilingual pre-training benefits our task or if it is a detriment (see Section 4.1).

Model	Citation	Huggingface link	Language	# Parameters
bert-base-cased	Devlin et al. (2018)	https://huggingface.co/bert-base-cased	English	110M
bert-large-cased	Devlin et al. (2018)	https://huggingface.co/bert-large-cased	English	340M
roberta-base	Liu et al. (2019)	https://huggingface.co/roberta-base	English	123M
gpt2	Radford et al. (2019)	https://huggingface.co/gpt2	English	1.5B
dbmdz/bert-base-turkish-cased		https://huggingface.co/dbmdz/bert-base-turkish-cased	Turkish	110M
bert-base-chinese		https://huggingface.co/bert-base-chinese	Chinese	110M
bert-base-multilingual-cased	Devlin et al. (2018)	https://huggingface.co/bert-base-multilingual-cased	104 languages	110M

Table 3.1: Huggingface transformer models

The number of parameters is considered a good measure of how much generalization a model can achieve, but also how much storage size and computer resources are required to train or fine-tune it. As shown in Table 3.1, the GPT2 model is an order of magnitude greater than the BERT base models (1.5B parameters versus 110M to 340M). It was used specifically to evaluate the influence of its size on the performance for out task.

3.1.2 PyTorch

All our models (see Section 3.2) use various tools provided by PyTorch[2]. PyTorch is an open source Python machine learning library that makes the creation and training of deep learning models on the GPU simple. It performs dynamic tensor computations with automatic differentiation. In particular, we have used PyTorch to implement LSTMs, GRUs, and Linear networks, as well as functions such as Dropout, Dataset, DataLoader, and CrossEntropyLoss.

The CRF[3] used is one that is implemented using PyTorch. This was used because it was easy to insert in whatever model that required it and it is open source. The PyTorch-CRF documentation states that the implementation is based on the one in the AllenNLP[4] CRF module found on GitHub[5].

[2] https://pytorch.org/

[3] https://pytorch-crf.readthedocs.io/en/stable/

[4] https://allenai.org/allennlp

[5] https://github.com/allenai/allennlp/blob/master/allennlp/modules/conditional_random_field.py

3.1.3 Training Set-up

All the models developed were fine-tuned for a maximum of 40 epochs using early stopping with a patience of 20 epochs. The models that are monolingual were trained on the appropriate languages; for example, `bert-base-cased` (English) was fined-tuned on the PDTB only. On the other hand multilingual models such as `bert-base-multilingual-cased`, were fine-tuned in a multilingual way using all three language corpora.

3.2 Models

Figure 3.1 shows the general architecture of the models we have developed. Each model is composed of a pre-trained embedding provided by Huggingface, and a classification head, which predicts the label for each token. By experimenting with different embeddings and classifiers we have created 6 specific models. Sections 3.2.1 to 3.2.6 will describe these in detail.

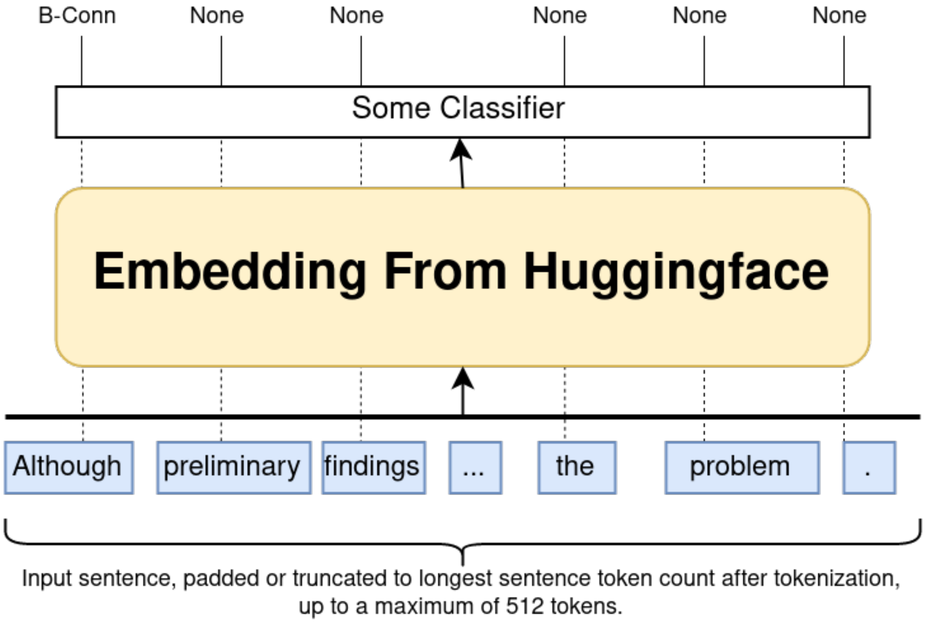

B-Conn
None
None
None
None
None
Some Classifier
Embedding From Huggingface
Although
preliminary
findings
...
the
problem
.
Input sentence, padded or truncated to longest sentence token count after tokenization, up to a maximum of 512 tokens.

Figure 3.1: General architecture of all models.

3.2.1 Model 1: BERT + Linear Layer

Model 1 is a standard BERT model. It uses a BERT embedding and a Linear layer classifier. Figure 3.2 shows a diagram of this model. The input is tokenized using the BERT word piece tokenizer (see Section 2.1.6); these tokens are then passed to the BERT embedding, the result of which is then passed to a feed forward neural network made up of a linear layer that outputs 3 values, one for each class, and the most probable class is selected. For the embeddings, we experimented with `bert-base-cased` for English, `dbmdz/bert-base-turkish-cased` for Turkish, `bert-base-chinese` for Chinese, and `bert-base-multilingual-cased` with all languages are trained together.

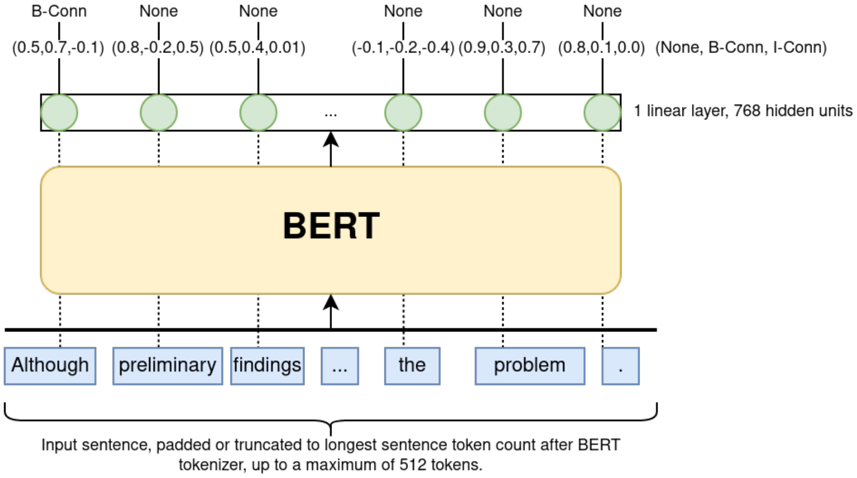


Figure 3.2: Overview of Model 1: BERT-base DC annotation model for English, Turkish, and/or Chinese

As shown in Figure 3.2, the linear layer consist of 1 layer with 768 hidden units.

3.2.2 Model 2: BERT + BiLSTM + Linear Layer

For the second model, we wanted to investigate how LSTMs would affect the fine-tuning of BERT. The model adds 2 layers of bi-directional LSTMs between the BERT embedding and the linear output layer. The size of the hidden layers of the BiLSTMs was set to 64, due to a limitation in computational resources at the time. This model architecture is similar to the one of the best performing models at DISRPT 2019 known as ToNy (Muller et al., 2019) for this particular task; however, our model was allowed to train longer, had an additional BiLSTM layer, and had fewer hidden units in the BiLSTMs. Figure 3.3 shows a diagram of the implemented model.

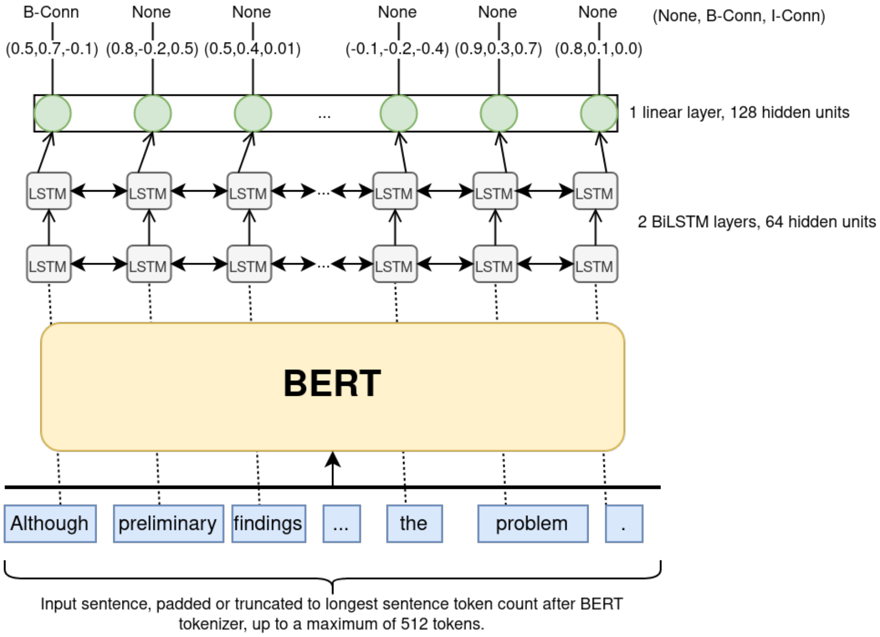

B-Conn
None
None
None
None
None
(None, B-Conn, I-Conn)
(0.5,0.7,-0.1)
(0.8,-0.2,0.5)
(0.5,0.4,0.01)
(-0.1,-0.2,-0.4)
(0.9,0.3,0.7)
(0.8,0.1,0.0)
...
1 linear layer, 128 hidden units
LSTM
2 BiLSTM layers, 64 hidden units
BERT
Although
preliminary
findings
...
the
problem
.
Input sentence, padded or truncated to longest sentence token count after BERT tokenizer, up to a maximum of 512 tokens.

Figure 3.3: Overview of Model 2: BERT-base with BiLSTM DC annotation model for English, Turkish, and/or Chinese

3.2.3 Model 3: BERT + BiGRU + Linear Layer

The third model is similar to the model with the LSTMs (Section 3.2.2) but the LSTMs are replaced by 2 layers of bi-directional GRUs, which also have 64 hidden units. Figure 3.4 shows a diagram of the implemented model. Because GRUs have fewer parameters than the LSTMs, we wanted to experiment with them to validate whether the performance would suffer and whether the performance of the task depends on how well the RNN can learn this task.

3.2.4 Model 4: BERT + Linear Layer + CRF

The fourth model considers the task as a sequence labelling task as opposed to an individual classifications. Hence, we wanted to evaluate the use of a CRF as the last layer of the model. Figure 3.5 shows a diagram of the implemented model. Similarly to the previous models, model 4 also uses BERT embeddings (`bert-base-cased` or `bert-large-cased` for English, `bert-base-chinese` for Chinese, `dbmdz/bert-base-turkish-cased` for Turkish, and `bert-base-multilingual-case`

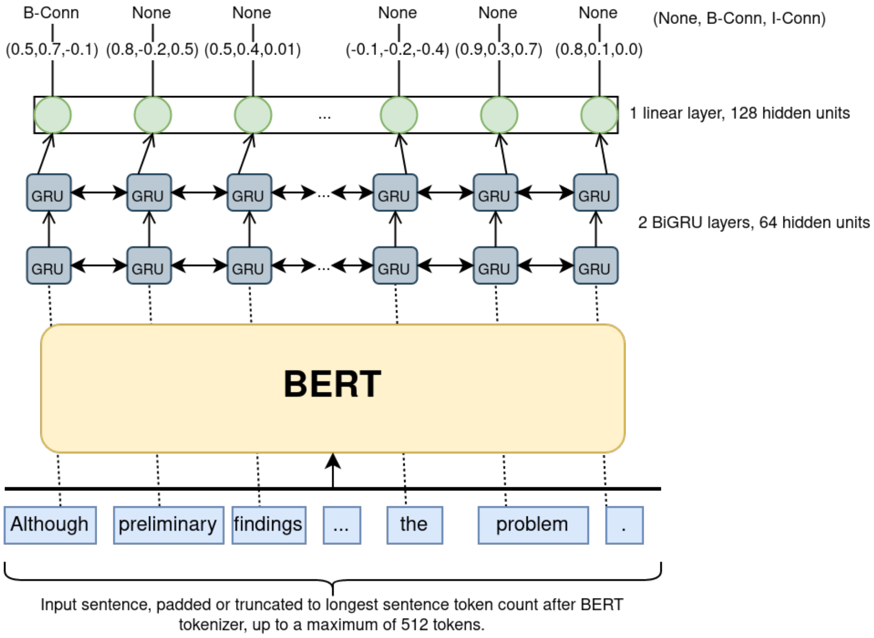

B-Conn
None
None
None
None
None
(None, B-Conn, I-Conn)
(0.5,0.7,-0.1)
(0.8,-0.2,0.5)
(0.5,0.4,0.01)
(-0.1,-0.2,-0.4)
(0.9,0.3,0.7)
(0.8,0.1,0.0)
1 linear layer, 128 hidden units
GRU
2 BiGRU layers, 64 hidden units
BERT
Although
preliminary
findings
...
the
problem
.
Input sentence, padded or truncated to longest sentence token count after BERT tokenizer, up to a maximum of 512 tokens.

Figure 3.4: Overview of Model 3: BERT-base with BiGRU DC annotation model for English, Turkish, and/or Chinese

for all) and the output is then sent to a linear layer that produces a score for each of the 3 labels (`B-Conn`, `I-Conn`, and `None`). In model 4, these scores are then fed to a conditional random field (CRF) (see Section 2.1.3) that produces the most likely final tags for each word given the whole sentence into account.

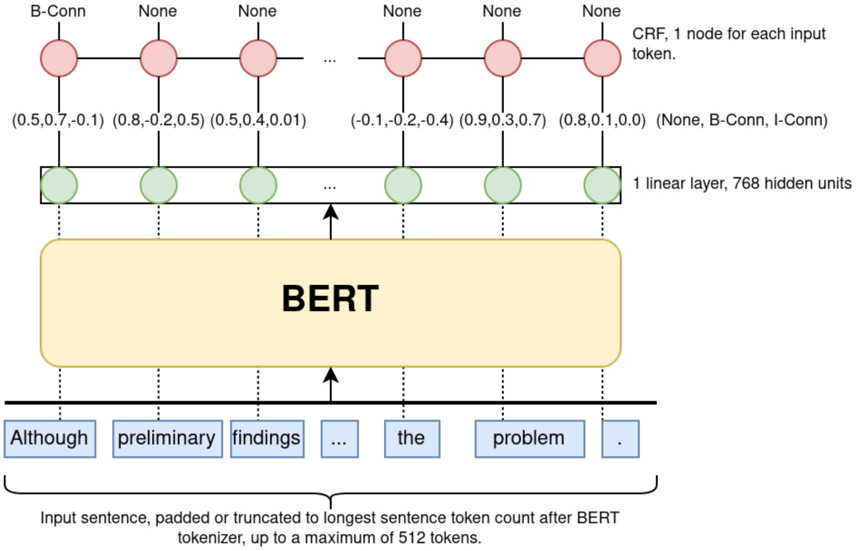

B-Conn
None
None
None
None
None
CRF, 1 node for each input token.
(0.5,0.7,-0.1)
(0.8,-0.2,0.5)
(0.5,0.4,0.01)
(-0.1,-0.2,-0.4)
(0.9,0.3,0.7)
(0.8,0.1,0.0)
(None, B-Conn, I-Conn)
1 linear layer, 768 hidden units
BERT
Although
preliminary
findings
...
the
problem
.
Input sentence, padded or truncated to longest sentence token count after BERT tokenizer, up to a maximum of 512 tokens.

Figure 3.5: Overview of Model 4: BERT-base with CRF output DC annotation model for English, Turkish, and/or Chinese

3.2.5 Model 5: RoBERTa + Linear Layer + CRF

The fifth model is similar to model 4, but replaces the BERT embedding with RoBERTa. However recall from Section 2.1.6 that RoBERTa uses the byte-pair encoding algorithm for tokenization; hence this can have an effect on the output as well. This model was experimented with to evaluate how a model that has slightly more parameters compared to BERT performs on this task. Figure 3.6 shows a diagram of the implemented model.

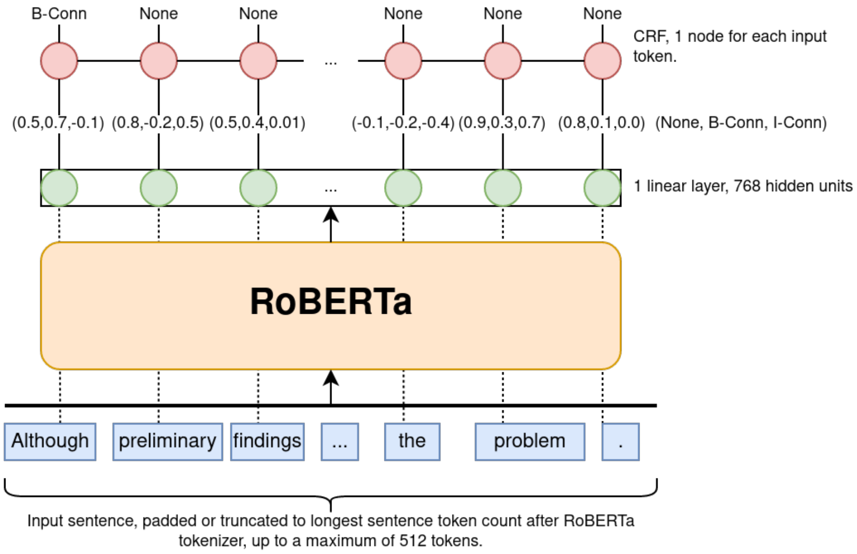

B-Conn
None
None
None
None
None
CRF, 1 node for each input token.
...
(0.5,0.7,-0.1)
(0.8,-0.2,0.5)
(0.5,0.4,0.01)
(-0.1,-0.2,-0.4)
(0.9,0.3,0.7)
(0.8,0.1,0.0)
(None, B-Conn, I-Conn)
...
1 linear layer, 768 hidden units
RoBERTa
Although
preliminary
findings
...
the
problem
.
Input sentence, padded or truncated to longest sentence token count after RoBERTa tokenizer, up to a maximum of 512 tokens.

Figure 3.6: Overview of Model 5: RoBERTa-base with CRF output DC annotation model for English

Due to the unavailability of RoBERTa embeddings for Turkish and Chinese (see Table 3.1) this model was only used for English.

3.2.6 Model 6: GPT2 + Linear Layer + CRF

The last model that we experimented with replaces the BERT embedding of model 4 with a GPT2 transformer embedding. As described in Section 2.1.6, the tokenization is done using GPT2 byte-pair encoding, like the RoBERTa model. The GPT2 model contains learning parameters that are an order of magnitude greater than the BERT model (1.5B versus 110M); therefore, we expected this model to be able to generalize and perform better on the test set. Figure 3.7 shows a diagram of the implemented model. Similarly to model 5, this model is only available for English and is only trained on the PDTB.

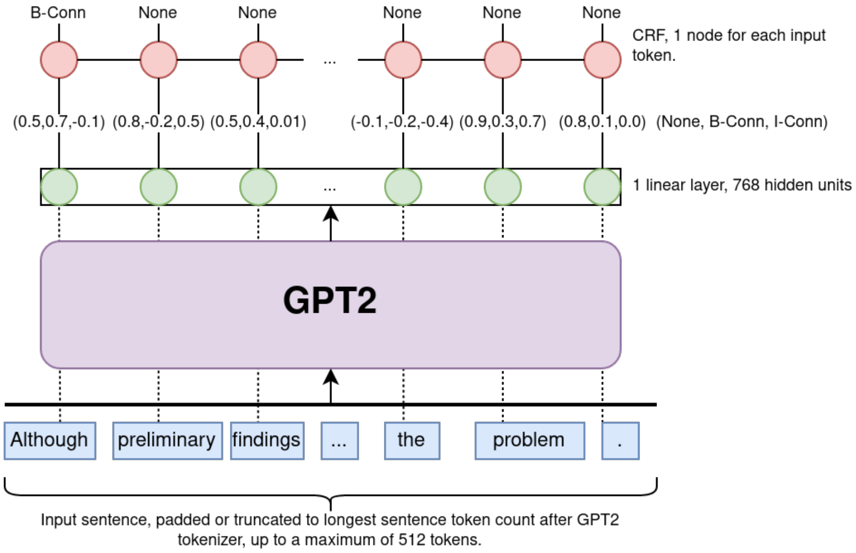

B-Conn
None
None
None
None
None
CRF, 1 node for each input token.
...
(0.5,0.7,-0.1)
(0.8,-0.2,0.5)
(0.5,0.4,0.01)
(-0.1,-0.2,-0.4)
(0.9,0.3,0.7)
(0.8,0.1,0.0)
(None, B-Conn, I-Conn)
1 linear layer, 768 hidden units
GPT2
Although
preliminary
findings
...
the
problem
.
Input sentence, padded or truncated to longest sentence token count after GPT2 tokenizer, up to a maximum of 512 tokens.

Figure 3.7: Overview of Model 6: GPT2-base with CRF output DC annotation model for English

All models above were experimented with in conjunction with corpus augmentation (see Section 3.3) and corpus reduction (see Section 3.4)

3.3 Corpus Augmentation

Data augmentation has been shown to increase performance in many NLP tasks (Bentivogli and Pianta, 2005; Tiedemann, 2015; Laali and Kosseim, 2017). Since DCs are semantic and rhetorical in nature, it is often assumed that discourse annotations can be projected from one language to another through word alignment. Therefore creating synthetic corpora with discourse connective annotations from resource rich languages to lower resource languages could help improve the performance of discourse connective identification models for these low resource languages. Given that the Chinese CDTB only contains 2049 training instances (see Table 2.4), we explored the use of data augmentation for this language. We also applied the data augmentation to Turkish to see if additional data would benefit the task.

We developed two methods for data augmentation: Annotation Projection (see Section 3.3.1.1) and Annotation Agreement (see Section 3.3.1.2). The first step in both approaches is to collect a list of DCs in both the low and resource rich languages. These lists contain words annotated with

`B-Conn` and a phrases (i.e. containing at least one `I-Conn`) and is extracted from the training sets of each language. These lists will be used to determine if no DCs are present in a given sentence or if the DC is present but not annotated by a model (i.e. in non-discourse usage). Once we have a list of DCs in each language, we used then for both methods: annotation projection and annotation agreement.

3.3.1 Methods

3.3.1.1 Annotation Projection

Annotation projection assigns an annotation from a source language word onto its aligned target language word, given a word aligned corpus. For the dataset used at DISRPT 2021, annotation projection involves the projection of the tags `B-Conn` and `I-Conn` from an English source dataset onto a parallel word aligned target dataset. This is done by training DC identification models for a source language (English) and a target language (Chinese or Turkish), then applying the trained models to identify DCs in a parallel word aligned corpus. It is important to note that the projection implemented in this research will only project annotations from the source language to the target language if the target language instance does not contain any DC annotations. This is done in order to capture DC annotations which the target language model was unable to label but which the source language model did label. However, the correctness of the label in the source language is undetermined, and this may lead to projecting incorrectly labeled DCs onto the target language. This is a particular weakness of this method. Figures 3.8, 3.9, and 3.10 show how differently annotated English and Chinese sentences will produce different outcomes. The algorithm for projection is presented in Listing 3.1 and the method is detailed below. Example 1 in Figure 3.8, shows a DC from English being projected onto the aligned Chinese phrases and being added to the synthetic corpus. Example 2 in Figure 3.9, shows a sentence being added in the resulting synthetic corpus that has no annotated DCs. This is because as Listing 3.1 shows, if the source sentence contains a connective the target sentence is added. Example 3 in Figure 3.10, shows a sentence that is not added because, as Listing 3.1 shows, if the target already has annotations, the sentence is not added.

Listing 3.1: Projection Algorithm

```
SET source_sents TO [[sent]]
SET source_sent_conns TO [[conn]]
SET target_sents TO [[sent]]
SET target_sent_conns TO [[conn]]
SET alignments_target_to_source TO [[alignment]]
SET target_connectives TO {read_json_file}
SET ndu_counter TO 0

FOR sent_idx, alignment IN enumerate(alignments_target_to_source):
    IF len(alignment) <= 3:
        continue
    SET source_conn to source_sent_conns[idx]
    SET target_conn to [None] * len(target_sent_conns[idx])
    SET target_sent to target_sents[idx]

    SET all_none_target TO [i is None FOR i IN target_sent_conns[idx]]
    SET all_none_source TO [i is None FOR i IN source_conn]
    SET error TO False
    IF all(all_none_target) and NOT all(all_none_source):
        FOR item in alignment:
            FOR target_idx, source_idx IN item.items():
                TRY:
                    IF alignment valid:
                        SET target_conn[target_idx] TO source_conn[source_idx]
                EXCEPT:
                    SET error TO True
                    continue
    ELSE:
        continue
    IF error:
        continue

    SET source_conn TO CLEAN_CONN_FORM(source_conn)
    SET all_none TO [i is None FOR i IN source_conn]
    IF all(all_none):
        IF NOT CHECK_IF_NDU(source_conn):
            continue
        ELSE:
            ndu_counter += 1
            IF ndu_counter % 2 EQUALS 0:
                continue

    WRITE_TO_CONLLU_FILE(source_conn)
```

For each alignment:

(1) Get all the tags in the English sentence and target language sentence.

(2) Check if:

- All the tags in target language sentence are None i.e. there is no DC in the target language, and
- All tags in the English sentence are not None i.e. at least one English token is marked as a DC.

(3) If (2) is true, iterate through the word alignments for this sentence, and project the tag found

on the aligned English word to the aligned target language word. If a target language word is aligned to multiple English words (i.e. a 1:n word alignment) then project only the tag of the last aligned word to the target language word.

(4) Re-label the target sentence to make sure all DCs start with B-conn and not I-Conn.

(5) Drop all target sentences that contain no potential DC – i.e. no word in the sentence is part of the language specific list of DCs. (see Section 3.3)

(6) Drop 50% of the sentences with at least one potential DC marked as non discourse usage – i.e. at least one word in the sentence is part of the language specific list of DCs but is labeled as None. This makes the dataset more balanced, otherwise the NDUs might drown out the DUs.

(7) Add the target sentence to the new synthetic corpus.

3.3.1.2 Annotation Agreement

Similarly to the annotation projection method, agreement requires DC annotation models trained on a source language and target language and applies these models to a word aligned parallel corpus. Once the DCs have been identified in both languages, the models are considered to be in agreement if the annotations in the target language match the annotations in the aligned words in the source language. This creates a dataset with high precision, as the ensemble of two models needs to agree on the annotation. In our experiments, the matching criteria is strict on matching the None tags from a source language to a target language, i.e. all aligned words that are None in the source need to be None in the target. Whereas for the DCs, B-Conn or I-Conn are considered to match if either tag in the source language is aligned with any number of words tagged as either B-Conn or I-Conn. If many source words are aligned with one target word (i.e. a n:1 alignment), the alignment is considered a match only if the source words and the target word are all None or they all contain DC annotations (B-Conn or I-Conn). To clearly show how agreement works, Figures 3.11 to 3.16 show examples of agreement between English (source) and Chinese (target). Note that Figure 3.12 does have agreement between the source and target; however, it may not be

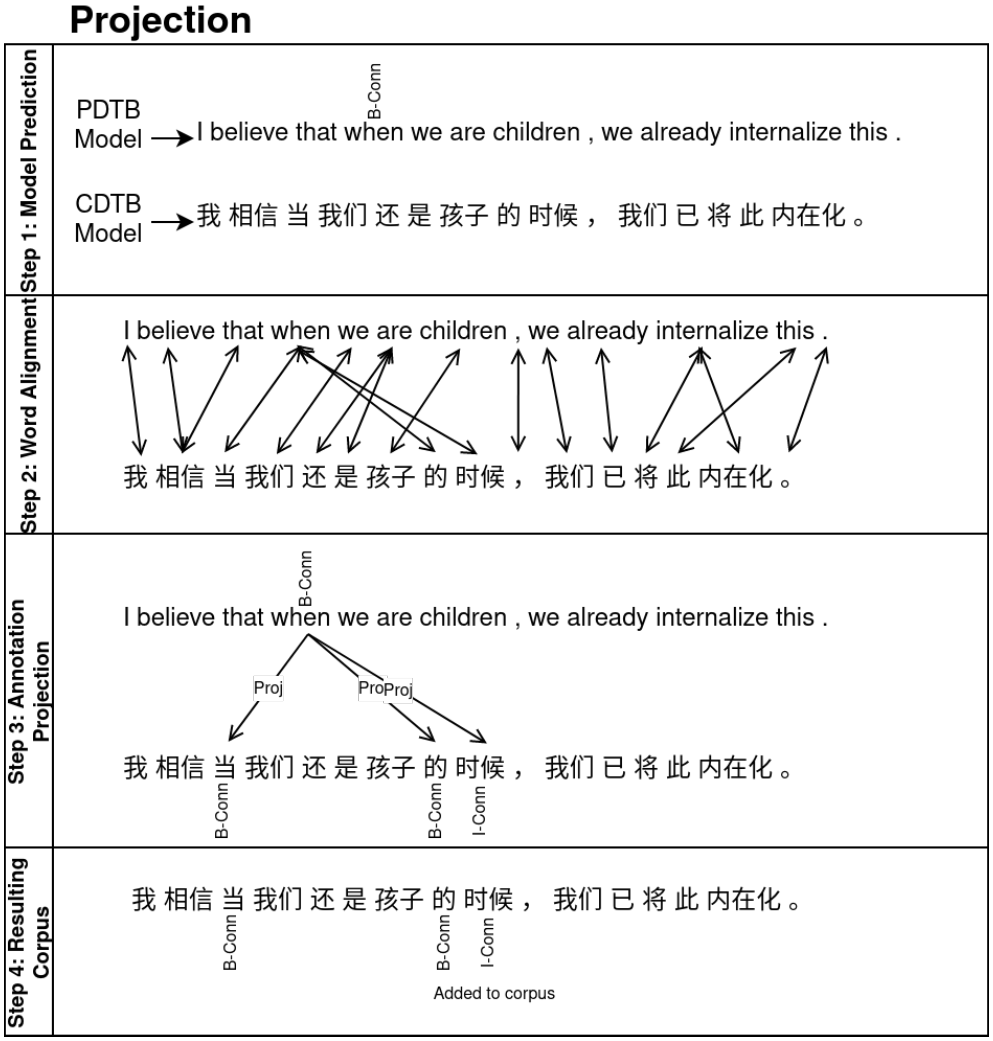

Projection
Step 1: Model Prediction
PDTB Model
B-Conn
I believe that when we are children , we already internalize this .
CDTB Model
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 2: Word Alignment
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 3: Annotation Projection
B-Conn
I believe that when we are children , we already internalize this .
Proj
ProProj
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
B-Conn
I-Conn
Step 4: Resulting Corpus
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
B-Conn
I-Conn
Added to corpus

Figure 3.8: Example 1: Annotation projection of an English discourse connective onto a Chinese sentence. The DC *when* is aligned to three Chinese words that are not tagged. These three words are tagged by the projection and the sentence is added to the synthetic corpus.

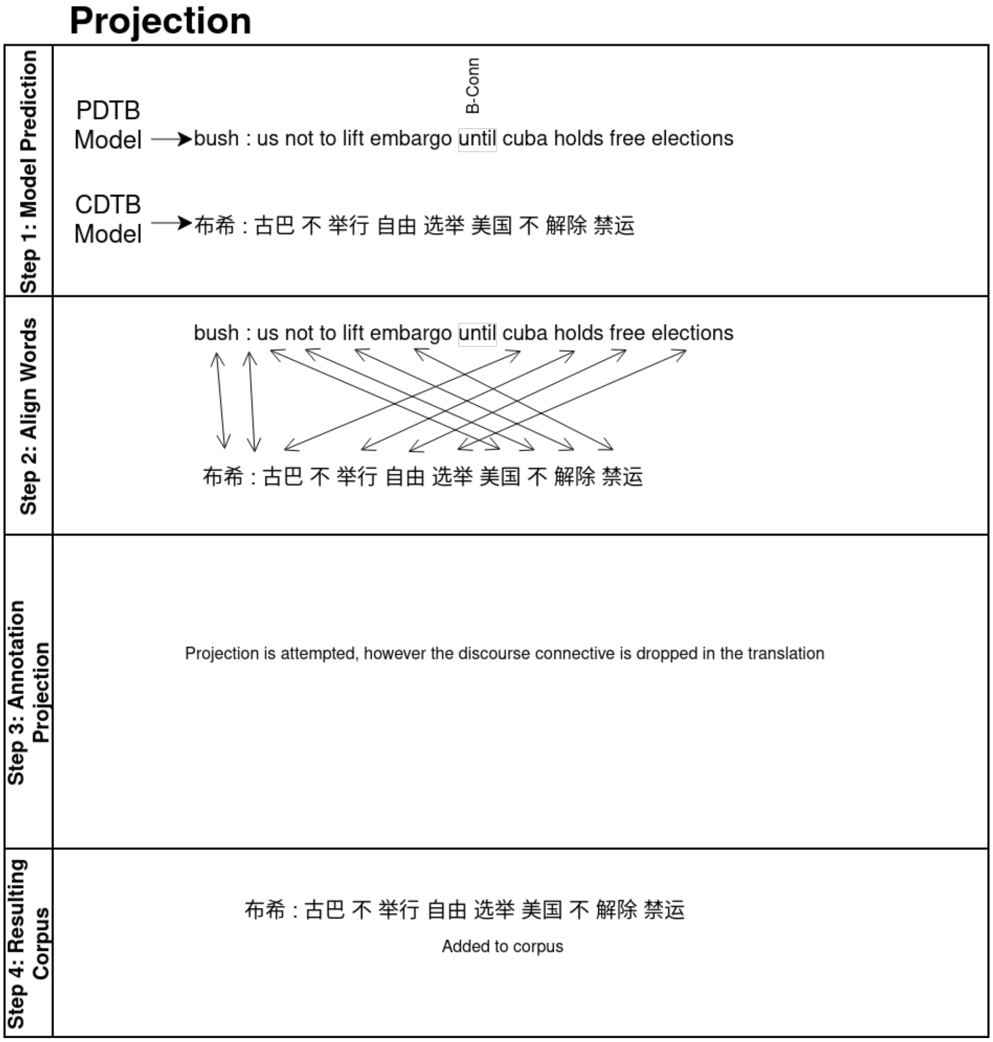

Projection
Step 1: Model Prediction
B-Conn
PDTB Model
bush : us not to lift embargo until cuba holds free elections
CDTB Model
布希 : 古巴 不 举行 自由 选举 美国 不 解除 禁运
Step 2: Align Words
bush : us not to lift embargo until cuba holds free elections
布希 : 古巴 不 举行 自由 选举 美国 不 解除 禁运
Step 3: Annotation Projection
Projection is attempted, however the discourse connective is dropped in the translation
Step 4: Resulting Corpus
布希 : 古巴 不 举行 自由 选举 美国 不 解除 禁运
Added to corpus

Figure 3.9: Example 2: Annotation projection of an English discourse connective dropped in Chinese. The DC *until* is dropped in the translation (i.e. not aligned), so the English annotation is not projected but the Chinese sentence is added to the synthetic corpus.

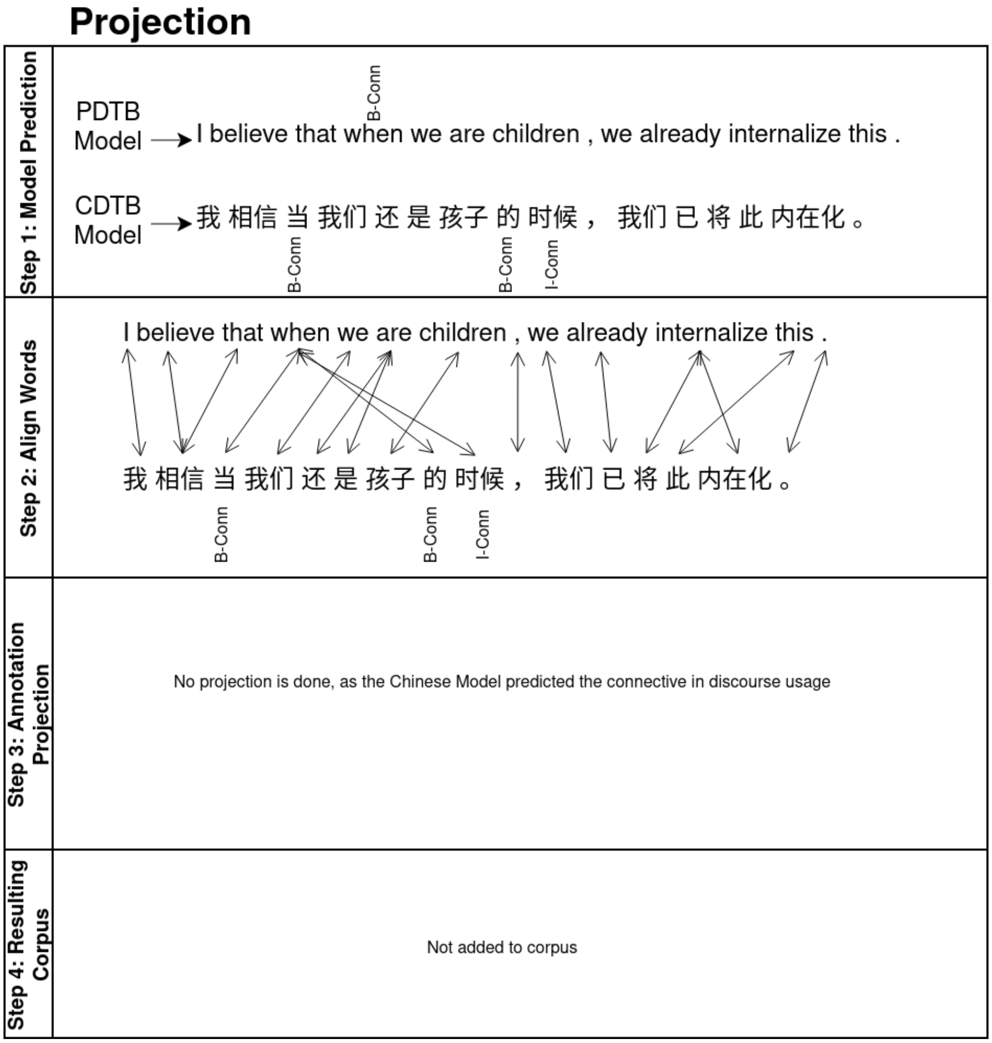

Projection
Step 1: Model Prediction
PDTB Model
I believe that when we are children , we already internalize this .
B-Conn
CDTB Model
我 相信 当 我们 还 是 孩子 的 时候， 我们 已 将 此 内在化。
B-Conn
B-Conn
I-Conn
Step 2: Align Words
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候， 我们 已 将 此 内在化。
B-Conn
B-Conn
I-Conn
Step 3: Annotation Projection
No projection is done, as the Chinese Model predicted the connective in discourse usage
Step 4: Resulting Corpus
Not added to corpus

Figure 3.10: Example 3: Annotation projection. The Chinese sentence already has annotated DCs, so the English DC annotation is not projected and the Chinese sentence is not added to the synthetic corpus.

added to the corpus because of steps 5 and 6. The algorithm is presented in Listing 3.2 and the method is detailed below.

Listing 3.2: Agreement Algorithm

```
SET source_sents TO [[sent]]
SET source_sent_conns TO [[conn]]
SET target_sents TO [[sent]]
SET target_sent_conns TO [[conn]]
SET alignments_target_to_source TO [[alignment]]
SET target_connectives TO {read_json_file}
SET ndu_counter TO 0

FOR sent_idx, alignment IN enumerate(alignments_target_to_source):
    IF len(alignment) <= 3:
        continue
    SET source_conn to source_sent_conns[idx]
    SET target_conn to target_sent_conns[idx]
    SET target_sent to target_sents[idx]

    SET all_equal TO []
    FOR item IN alignment:
        SET error TO False
        FOR target_idx, source_idx IN item.items():
            TRY:
                IF alignment valid:
                    IF (source_conn[source_idx] IS None AND target_conn[
                        target_idx] IS None) OR (source_conn[source_idx] IS NOT
                        None AND target_conn[target_idx] IS NOT None):
                        all_equal.append(True)
                ELSE:
                    all_equal.append(False)
            EXCEPT:
                all_equal.append(False)
                SET error TO True
                continue
        IF error:
            continue
    IF NOT all(all_equal):
        continue

    SET source_conn TO CLEAN_CONN_FORM(source_conn)
    SET all_none TO [i is None FOR i IN source_conn]
    IF all(all_none):
        IF NOT CHECK_IF_NDU(source_conn):
            continue
        ELSE:
            ndu_counter += 1
            IF ndu_counter % 2 EQUALS 0:
                continue

    WRITE_TO_CONLLU_FILE(source_conn)
```

For each alignment:

(1) Get all the tags in the English sentence and the target language sentence.

(2) Iterate through the word alignment for this sentence. A word's tags are considered a match if:

- the aligned English word is tagged as None and the target language word is also tagged as None.
- or if the aligned English word is not None and the target language word is not None.

If a target language word is aligned to multiple English words (i.e. an n:1 alignment), each aligned English word is checked for the above matching condition.

(3) If all aligned words match the above condition, we proceed to step 4 below, otherwise we continue to the next sentence.

(4) Relabel the target sentence, to ensure that all DCs start with `B-conn` and not `I-Conn`.

(5) Drop all sentences that contain no potential DC – i.e. no word in the sentence is part of the language specific list of DCs. (see Section 3.3)

(6) Drop 50% of the sentences in the target language with at least one potential DC that is marked as non discourse usage – i.e. at least one word in the sentence is part of the language specific list of DCs but is labeled as None.

(7) Add the new target sentence to new synthetic corpus.

Example 3 in Figure 3.13, the sentence has a chance of being added to the synthetic corpus, as in Listing 3.2 shows that if a DC is in NDU form, it has a 50% chance of being added to the corpus. Examples 5 and 6 in Figures 3.15 and 3.16 respectively show sentences that are not added to the corpus because they do not have a complete match on the annotated DCs, as Listing 3.2 requires.

3.3.1.3 Projection union Agreement

The corpora created by the projection and the agreement methods are mutually exclusive, which means they can be combined together to create a new DC annotated corpus. We took the union of the datasets by simply training with both corpora at the same time.

3.3.2 Resulting Synthetic Corpora

3.3.2.1 Synthetic Chinese Corpus

As indicated in Section 3.3.1.1, to apply annotation projection and agreement, we needed word aligned corpora. To create synthetic corpora for Chinese, we used the Tsinghua alignment evaluation set version 2 (Liu and Sun, 2015; Liu et al., 2005), which contains 40,716 manually word

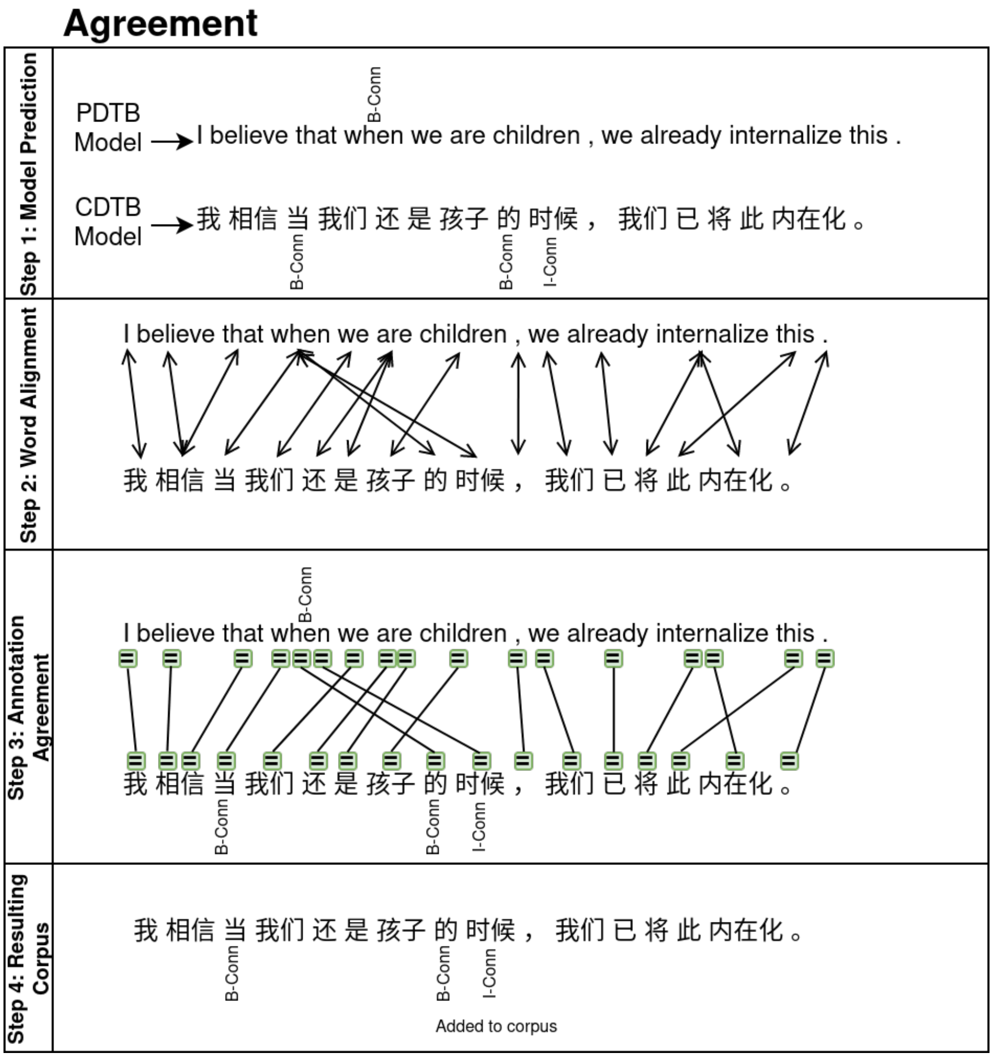

Agreement
Step 1: Model Prediction
PDTB Model
I believe that when we are children , we already internalize this .
B-Conn
CDTB Model
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
B-Conn
I-Conn
Step 2: Word Alignment
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 3: Annotation Agreement
B-Conn
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
B-Conn
I-Conn
Step 4: Resulting Corpus
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
B-Conn
I-Conn
Added to corpus

Figure 3.11: Example 1: Annotation agreement does match. The DC *when* and its aligned words are annotated as DCs. The annotations agree so the sentence is added to the synthetic corpus.

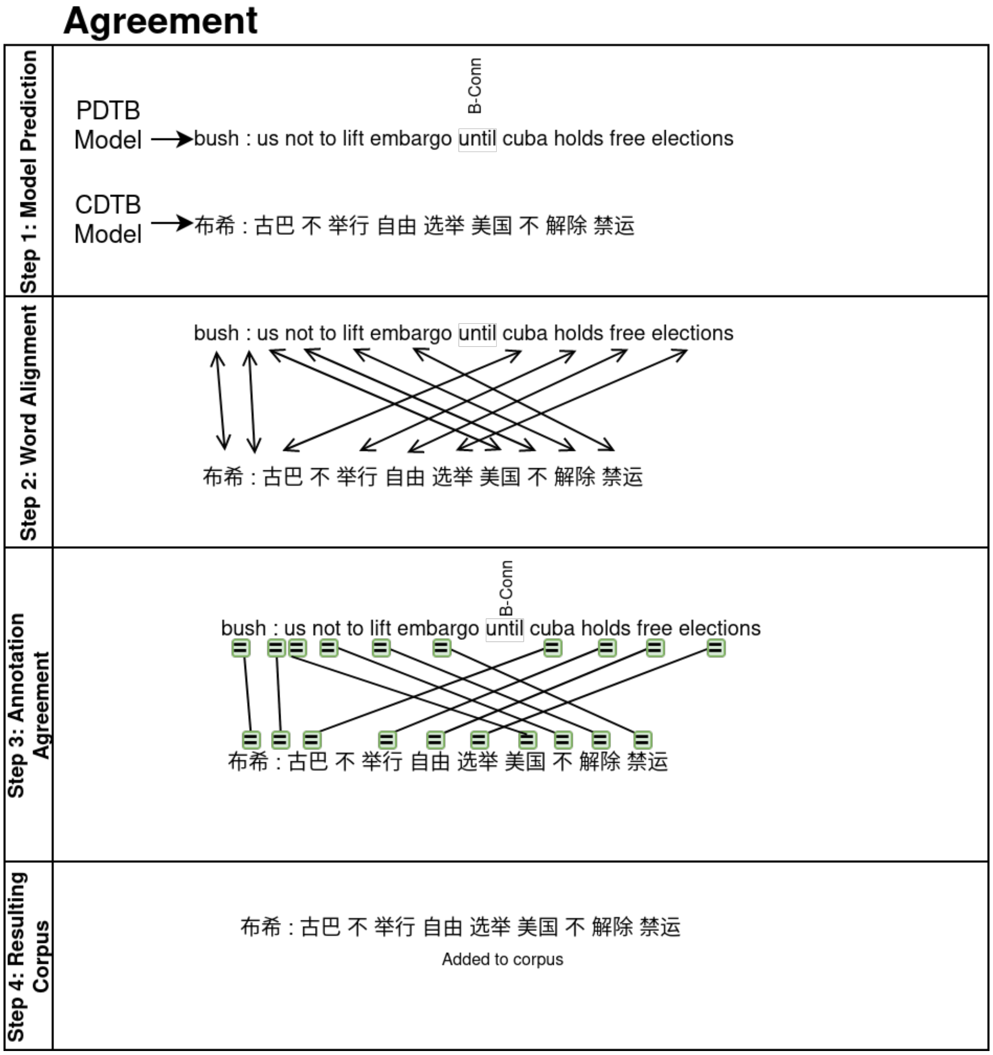

Agreement
Step 1: Model Prediction
PDTB Model
bush : us not to lift embargo until cuba holds free elections
B-Conn
CDTB Model
布希 : 古巴 不 举行 自由 选举 美国 不 解除 禁运
Step 2: Word Alignment
bush : us not to lift embargo until cuba holds free elections
布希 : 古巴 不 举行 自由 选举 美国 不 解除 禁运
Step 3: Annotation Agreement
B-Conn
bush : us not to lift embargo until cuba holds free elections
布希 : 古巴 不 举行 自由 选举 美国 不 解除 禁运
Step 4: Resulting Corpus
布希 : 古巴 不 举行 自由 选举 美国 不 解除 禁运
Added to corpus

Figure 3.12: Example 2: Annotation agreement does not match. English DC dropped in translation. The DC *until* is dropped in translation (i.e. not aligned), so the English annotation is not in agreement with the Chinese sentence, but the sentence could be added to the corpus if the sentence contains a NDU (i.e. labelled as `None`).

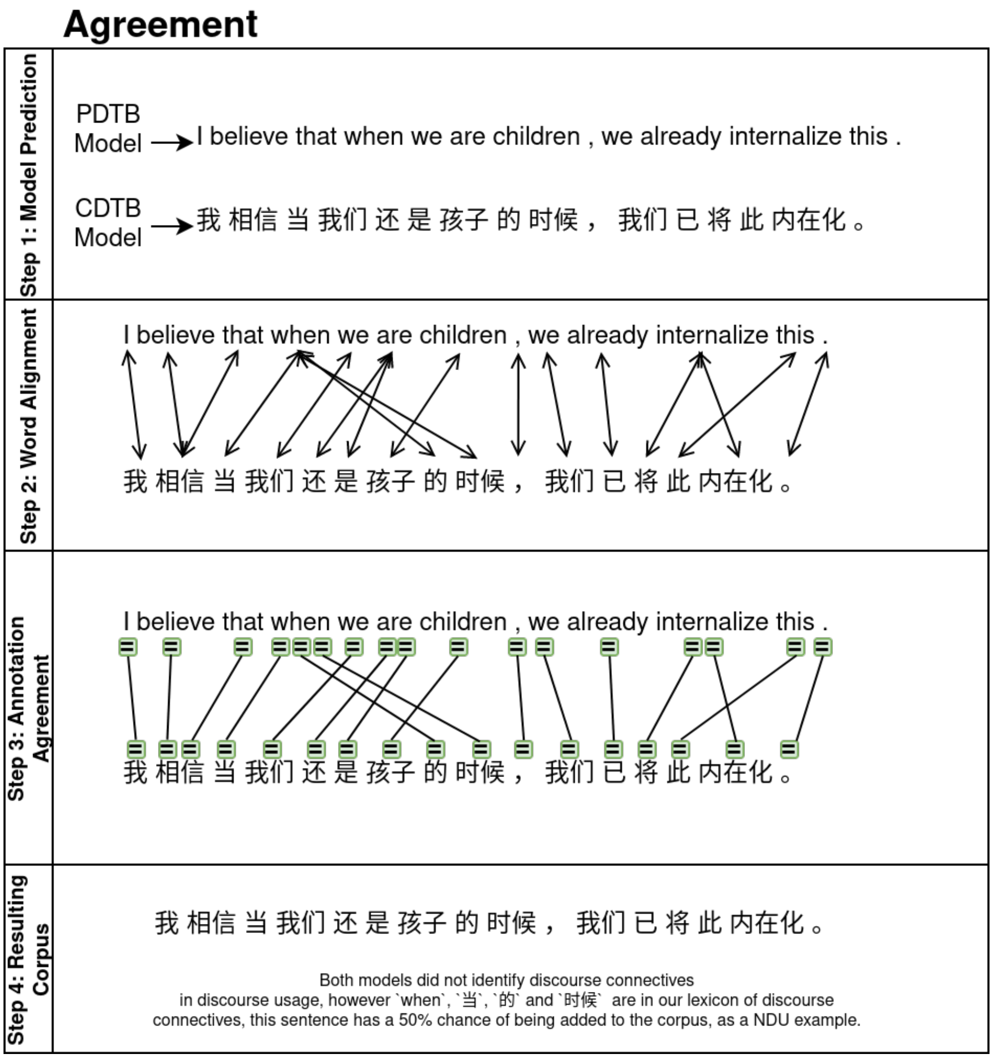

Agreement
Step 1: Model Prediction
PDTB Model
I believe that when we are children , we already internalize this .
CDTB Model
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 2: Word Alignment
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 3: Annotation Agreement
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 4: Resulting Corpus
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Both models did not identify discourse connectives
in discourse usage, however `when`, `当`, `的` and `时候` are in our lexicon of discourse
connectives, this sentence has a 50% chance of being added to the corpus, as a NDU example.

Figure 3.13: Example 3: Annotation agreement of English DC in NDU with a Chinese DC in NDU. The DC *when* and its aligned words are in NDU (i.e. labelled as `None`), meaning the annotation agrees and the sentence is added to the synthetic corpus because those words can be found in the list of DCs.

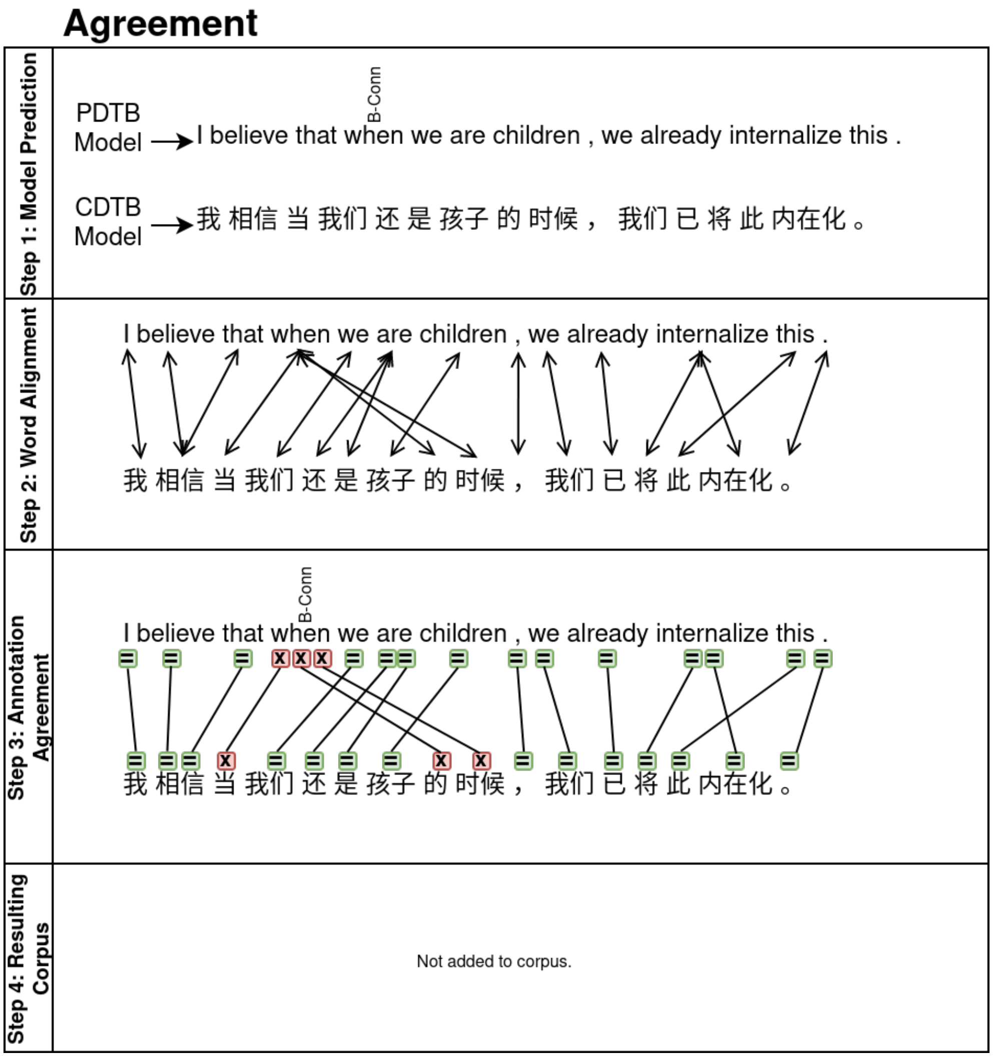

Agreement
Step 1: Model Prediction
PDTB Model
B-Conn
I believe that when we are children , we already internalize this .
CDTB Model
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 2: Word Alignment
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 3: Annotation Agreement
B-Conn
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 4: Resulting Corpus
Not added to corpus.

Figure 3.14: Example 4: Annotation are not in agreement, all of the Chinese DCs not annotated. The *when* is annotated but its aligned words are not. Therefore the Chinese sentences are not in agreement and the sentence is not added to the synthetic corpus.

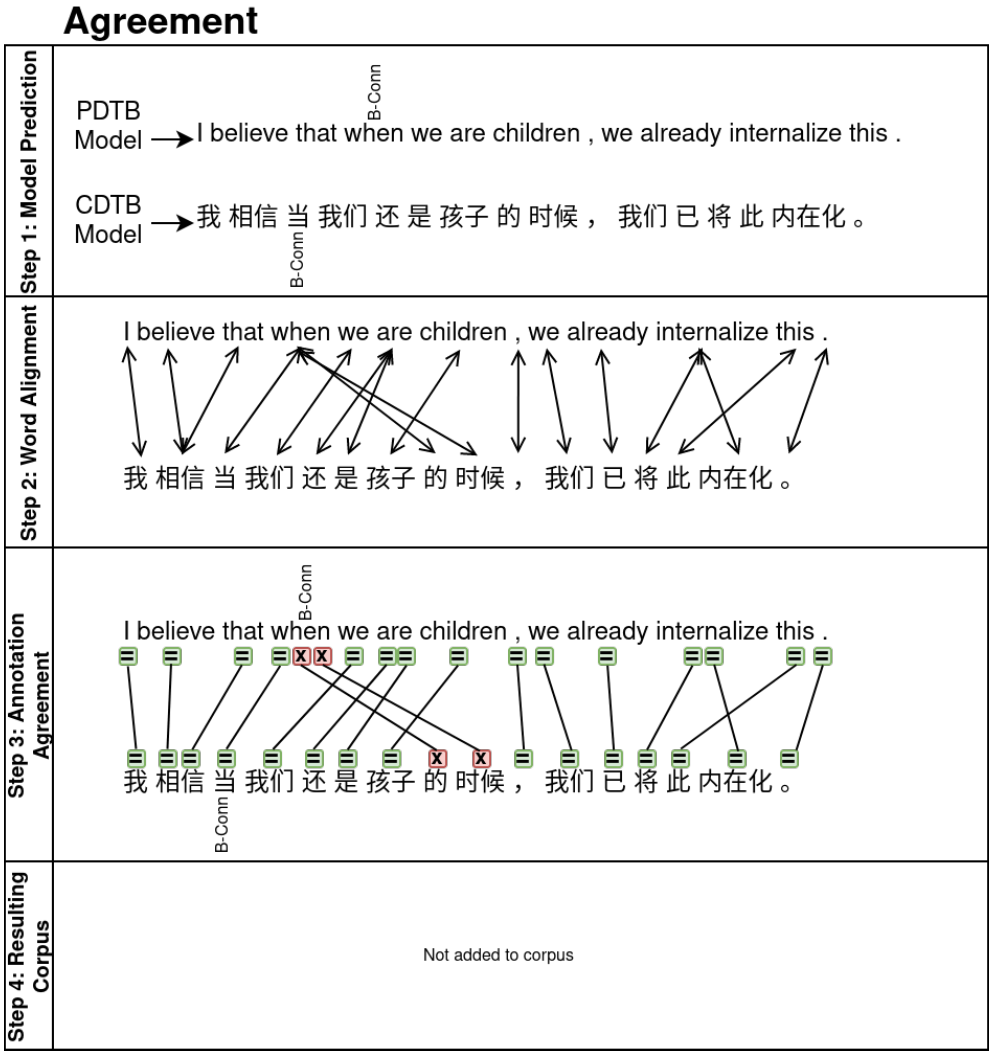

Agreement
Step 1: Model Prediction
PDTB Model
I believe that when we are children , we already internalize this .
B-Conn
CDTB Model
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
Step 2: Word Alignment
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 3: Annotation Agreement
B-Conn
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
Step 4: Resulting Corpus
Not added to corpus

Figure 3.15: Example 5: Annotation are not in agreement, part of the Chinese DC is not annotated. The *when* is annotated but its aligned words are only partly annotated. Therefore the sentences are not in agreement and the Chinese sentence is not added to the synthetic corpus.

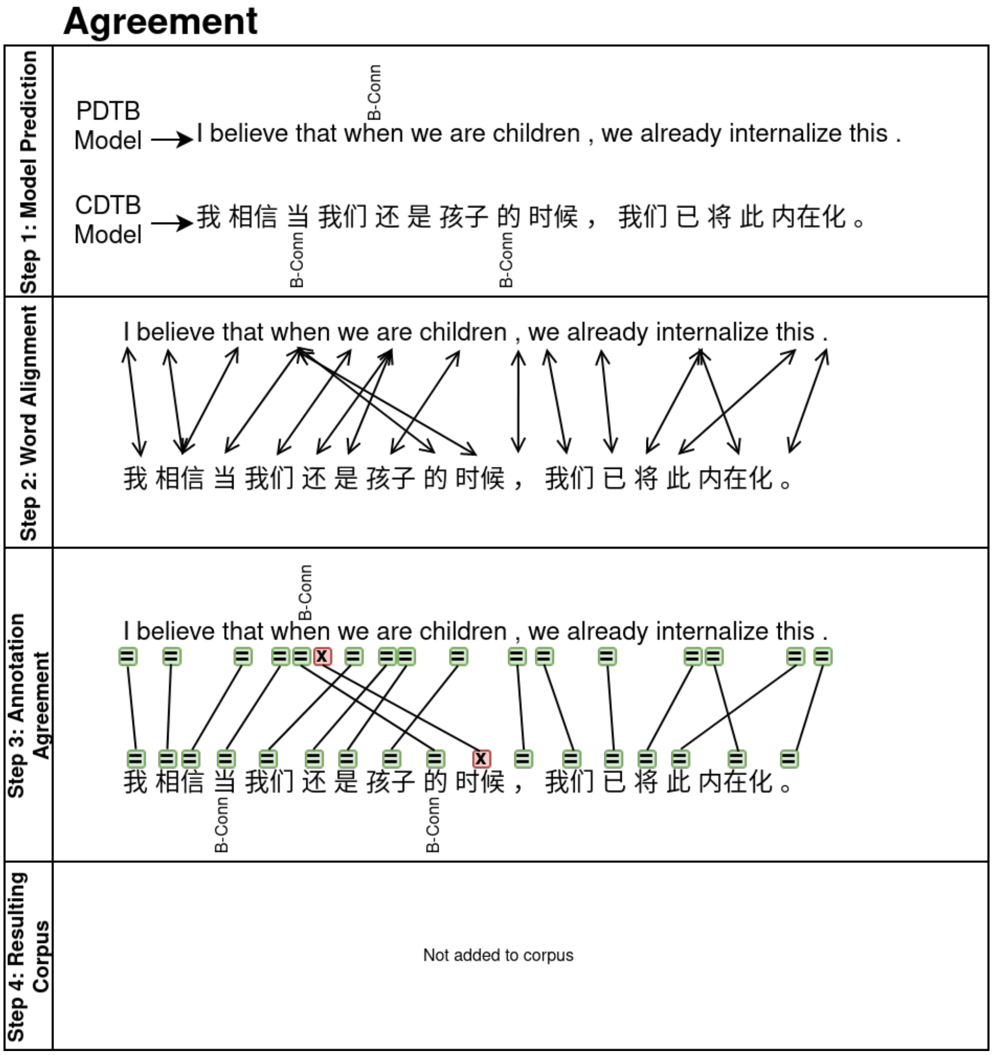

Agreement
Step 1: Model Prediction
PDTB Model
I believe that when we are children , we already internalize this .
B-Conn
CDTB Model
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
B-Conn
Step 2: Word Alignment
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
Step 3: Annotation Agreement
B-Conn
I believe that when we are children , we already internalize this .
我 相信 当 我们 还 是 孩子 的 时候 ， 我们 已 将 此 内在化 。
B-Conn
B-Conn
Step 4: Resulting Corpus
Not added to corpus

Figure 3.16: Example 6: Annotation are not in agreement, part of the Chinese DC is not annotated. The *when* is annotated but its aligned words are only partly annotated. Therefore the sentences are not in agreement and the Chinese sentence is not added to the synthetic corpus.

aligned sentences. This corpus was created to verify the performance of the Tsinghua statistical word alignment system. While the inter and intra annotator agreement of this gold dataset is unknown, this was the only large sentence and word aligned dataset we found at the time.

To create the synthetic Chinese DC annotated data sets, we first created a list of DCs for English and Chinese as defined in Section 3.3 by extracting the DCs labelled in the PDTB and the CDTB training sets. This lead to 1160 DCs for English and 195 DCs for Chinese. An English DC identification model was then trained on the PDTB training set and another model on the CDTB; the model architecture used was model 4, the BERT base with CRF output (Section 3.2.4). As the Tsinghua alignment eval set was already tokenized and included word alignment, therefore we were able to use it directly. The trained models were used to identify DCs in the parallel Tsinghua corpus; with the DC identified, the two methods (Projection Section 3.3.1.1, and Agreement Section 3.3.1.2) were applied to synthesize two new Chinese datasets with DC annotations. Table 3.2 shows the breakdown of the resulting corpus. The table shows that the agreement method (ZHO-AG) produced a synthetic data set with nearly 22k training instances which is a magnitude greater than the CDTB. Whereas, the projection method (ZHO-PJ) produced a synthetic data set with nearly 3k training instances, a little more than the CDTB training set.

3.3.2.2 Synthetic Turkish Corpus

The synthetic Turkish corpus is based on the SETimes (Tiedemann, 2012) English-Turkish parallel corpus, which contains 207,677 aligned sentences. SETimes is a parallel corpus of news articles in Balkan languages and English, though it is not word aligned. In order to align the words we used SimAlign (Jalili Sabet et al., 2020), which also provides the probability of the alignment. SimAlign can provide different alignment outputs based on 3 different algorithms: itermax, argmax, and match. Based on the results of Jalili Sabet et al. (2020), itermax seems to perform better than the other methods; hence, we used the itermax word alignments.

Similarly to Chinese, we extracted a list of Turkish and English DCs from the TDB and PDTB training set. This lead to a list of 1160 DCs for English and 277 DCs for Turkish. Then we trained model 4 the same BERT base with CRF output (Section 3.2.4) on the PDTB and TDB training sets. However, the SETimes corpus has no word alignment and is not tokenized. Therefore we used SimAlign (Jalili Sabet et al., 2020) to generate itermax alignment with probabilities and used Spacy (Honnibal et al., 2020) to tokenize the two sides of the parallel data. With the tokenized data, the DC identification models were applied. The SimAlign probabilities were used to keep only the sentences with an average word alignment probability 85% or greater, to ensure that error propagation is minimized. The two methods (Projection (Section 3.3.1.1) and Agreement (Section 3.3.1.2)) were applied resulting in two synthetic corpora (TUR-AG and TUR-PJ), the details of which can be seen in Table 3.2. The table shows that the agreement method (TUR-AG) produced a synthetic data set with nearly 28k training instances, which is larger than the TDB training set (25k instances). Whereas, the projection method (TUR-PJ) produced a synthetic data set of just over 4k training instances.

Synthetic Corpus	Language	# of Train Sentences	# tok `B-Conn` + `I-Conn`	% tok `B-Conn` + `I-Conn`
ZHO-AG	Chinese	21,934	21,774	4.645
ZHO-PJ	Chinese	2,848	1,404	2.312
TUR-AG	Turkish	27,827	6,537	1.254
TUR-PJ	Turkish	4,468	4,166	4.191

Table 3.2: Statistics of the Chinese and Turkish synthetic datasets

3.4 Corpus Reduction

In order to better evaluate the influence of the size of the training data for DC annotation, we also experimented with corpus reduction. To do so, we created random subsets of the PDTB, the TDB and the CDTB. The subsets are 75%, 50%, 25%, 10% and 5% of the original datasets for each language. These experiments were carried out to determine how much data is needed until the performance of identifying DCs is significantly affected. Additionally 5% of the PDTB and 10% of the TDB are comparable in size to the entire CDTB allowing us to determine if identifying DCs is a simpler task in one of the languages compared to the others.

In this chapter, we described the six models we developed to run various experiments for the automatic identification of DCs. We also described the tools and frameworks used to create our models. Each of these models uses a transformer as an embedding; and only BERT based models are available for Turkish and Chinese, while the other models, built only for English, were built to evaluate if BERT is sufficient for accomplishing this task or if additional parameters are of benefit. Additionally, in Section 3.3 we discussed and explained the methods we developed to modify the corpora by data augmentation using projection and agreement and by data reduction. In Chapter 4, each of these models and methods will be used and the performance on the development and test sets for the task of DC identification will be presented and analysed.

Chapter 4

Results and Analysis

In this chapter, the results of each models' performances on the DISRPT 2021 will be presented along with a general analysis of these results (see Section 4.1). Section 4.2 will then present a more comprehensive error analysis on the best performing models for English, Turkish and Chinese in the hope of understanding where the models can be improved.

4.1 DISRPT 2021 Results

For all models presented in Chapter 3, we ran several experiments varying the training corpora. Each experiment was run five times along with we recorded the average score and the standard deviation.

4.1.1 English Results

The English DC identification results are presented in Table 4.1. As the table shows, for most experiments, excluding the GPT2 model and data reduction, the lowest F-measure on the test set is 88.18 (model 1 on row 15) and the highest is 93.13 (model 4 on row 7); that is a difference of only 4.95 points. The model that achieves the best score is model 4 with the BERT large embedding. However, this model does require much more computational resources than model 4 that uses the BERT base embedding (row 1) or model 1 with the BERT base embedding (row 10) which lead to F-measures of 92.49 and 92.99 respectively. It would be interesting to empirical measure the energy consumption of training these models in order to measure the performance over their training cost. The results of Table 4.1 also show that the development set seems to be easier than the test set for the task of English DC identification, as the F-measures seen on the development set are always slightly higher than these of the test set.

Unsurprisingly the experiments with the subsets of the PDTB lead to strong results, as even with only 5% of the PDTB (row 6) there are still more training instances than the CDTB. These results indicate that the BERT transformer does benefit from having more data, but the performance improvement over the size of the data gives diminishing returns. As this type of annotated data is difficult and costly to create, finding a balance between the number of training instances and performance is important and might help guide the creation of such resources for other languages.

Recall that the GPT2 model (model 6 on row 9) has an order of magnitude more learning parameters than the BERT base model (model 4 on row 7), yet it performs rather poorly (F-measure of 81.20 ±1.17). This could be because the CRF output does not backpropagate good error signals to the GPT2 transformer model, or maybe this is not a task the GPT2 transformer does well in. More investigation is needed in order to determine what the problem is. Similarly, the RoBERTa model (model 5 on row 8) under-performs, possibly for similar reasons as the GPT2 model. Note that both transformers use the BPE algorithm for tokenization; while BERT uses Word Pieces (see Section 2.1.6). This may explain the superior preformance of BERT.

4.1.2 Turkish Results

Table 4.2 shows the performance of various models on the Turkish data set. The models have overall the best performance on the TDB test set for Turkish DC identification over DC identification for the other two languages; this is likely due to the simpler task of identifying `Explicit` DC and phrasal expressions (small subset of `AltLex`). The best F-measure attained is 94.42 by the BERT (`dbmdz/bert-base-turkish-cased`) model (model 1 on row 13), but it seems slightly less stable than model 4: BERT + CRF (model 4 on row 1) as it has a standard deviation of 0.37 compared to 0.31. Another thing to notice is that the development dataset seems to be a slightly more difficult task, as each model performs 1 or more points better on the test set than on the development set.

				PDTB Dev Set			PDTB Test Set		
Row	**Model #**	**Model**	**Training Dataset**	**Dev Precision**	**Dev Recall**	**Dev F-measure**	**Test Precision**	**Test Recall**	**Test F-measure**
1	Model 4	bert-base-cased + CRF	PDTB	94.00 (±0.64)	93.68 (±0.31)	93.84 (±0.46)	92.69 (±1.16)	92.31 (±0.41)	92.49 (±0.77)
2	Model 4	bert-base-cased + CRF	PDTB-75%	94.02 (±0.46)	92.93 (±0.75)	93.47 (±0.33)	92.99 (±0.56)	91.73 (±1.12)	92.35 (±0.48)
3	Model 4	bert-base-cased + CRF	PDTB-50%	93.28 (±0.60)	92.55 (±0.29)	92.91 (±0.33)	91.77 (±0.90)	91.61 (±0.56)	91.69 (±0.33)
4	Model 4	bert-base-cased + CRF	PDTB-25%	93.19 (±0.68)	92.49 (±0.50)	92.84 (±0.43)	92.04 (±0.88)	90.81 (±0.89)	91.41 (±0.13)
5	Model 4	bert-base-cased + CRF	PDTB-10%	92.47 (±0.68)	90.62 (±0.76)	91.53 (±0.37)	91.71 (±0.83)	89.66 (±0.67)	90.67 (±0.19)
6	Model 4	bert-base-cased + CRF	PDTB-05%	91.46 (±1.42)	89.89 (±0.67)	90.66 (±0.44)	90.66 (±1.33)	87.53 (±0.74)	89.06 (±0.45)
7	Model 4	bert-large-cased + CRF	PDTB	94.42 (±0.66)	93.62 (±0.39)	94.02 (±0.38)	93.46 (±0.95)	92.79 (±0.69)	**93.12 (±0.49)**
8	Model 5	RoBERTa + CRF	PDTB	92.88 (±1.21)	92.32 (±0.76)	92.59 (±0.58)	92.02 (±1.08)	90.97 (±0.83)	91.49 (±0.47)
9	Model 6	GPT2 + CRF	PDTB	83.28 (±1.69)	85.50 (±1.34)	84.36 (±0.95)	79.64 (±2.19)	82.86 (±1.21)	81.20 (±1.17)
10	Model 1	bert-base-cased	PDTB	94.34 (±0.57)	93.68 (±0.16)	94.01 (±0.22)	93.03 (±0.65)	92.95 (±0.42)	92.99 (±0.34)
11	Model 2	bert-base-cased + Bi-LSTM	PDTB	94.46 (±0.33)	94.14 (±0.53)	94.30 (±0.36)	92.68 (±0.62)	92.77 (±0.57)	92.72 (±0.52)
12	Model 3	bert-base-cased+ Bi-GRU	PDTB	94.38 (±0.73)	94.42 (±0.49)	94.40 (±0.35)	92.99 (±0.72)	92.56 (±0.65)	92.77 (±0.38)
13	Model 4	bert-base-multilingual-cased + CRF	PDTB + TDB + CDTB	93.22 (±1.07)	92.80 (±0.35)	93.01 (±0.50)	92.29 (±1.45)	91.63 (±0.54)	91.95 (±0.54)
14	Model 1	bert-base-multilingual-cased	PDTB + TDB + CDTB	93.01 (±0.58)	93.72 (±0.44)	93.36 (±0.20)	91.53 (±0.78)	92.64 (±0.33)	92.08 (±0.25)
15	Model 1	bert-base-multilingual-cased	PDTB-05% + TDB-10% + CDTB	89.23 (±0.46)	90.37 (±0.67)	89.79 (±0.49)	88.06 (±1.12)	88.31 (±0.64)	88.18 (±0.76)
16	—	baseline	PDTB	—	—	—	65.52	29.00	40.20

Table 4.1: Performance of various model configurations on the PDTB (English) development and test sets. Performance indicates the average score over five runs ± the standard deviation.

				TDB Dev Set			TDB Test Set		
Row	**Model #**	**Model**	**Training Dataset**	**Dev Precision**	**Dev Recall**	**Dev F-measure**	**Test Precision**	**Test Recall**	**Test F-measure**
1	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB	90.80 (±0.81)	94.72 (±0.95)	92.71 (±0.27)	93.19 (±1.08)	94.87 (±0.72)	94.01 (±0.31)
2	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB-75%	90.06 (±0.67)	94.55 (±0.75)	92.25 (±0.35)	92.45 (±0.92)	95.01 (±1.01)	93.70 (±0.32)
3	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB-50%	89.70 (±1.37)	93.77 (±1.15)	91.67 (±0.18)	90.73 (±1.42)	94.44 (±1.11)	92.54 (±0.41)
4	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB-25%	88.09 (±0.95)	92.47 (±0.86)	90.22 (±0.22)	90.48 (±0.83)	94.16 (±0.95)	92.28 (±0.49)
5	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB-10%	86.85 (±0.89)	89.95 (±1.09)	88.36 (±0.19)	88.98 (±0.72)	91.02 (±1.61)	89.98 (±0.67)
6	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB-05%	82.99 (±1.44)	84.74 (±1.58)	83.84 (±0.62)	85.84 (±1.27)	86.29 (±1.91)	86.04 (±0.41)
7	Model 4	dbmdz/bert-base-turkish-cased + CRF	TUR-AG	82.69 (±1.18)	85.23 (±1.11)	83.93 (±0.34)	88.22 (±1.87)	87.80 (±0.84)	87.99 (±0.54)
8	Model 4	dbmdz/bert-base-turkish-cased + CRF	TUR-PJ	32.86 (±2.12)	52.44 (±2.87)	40.30 (±1.13)	34.04 (±2.40)	52.39 (±3.71)	41.15 (±1.68)
9	Model 4	dbmdz/bert-base-turkish-cased + CRF	TUR-AG + TUR-PJ	62.78 (±1.18)	82.45 (±1.30)	71.28 (±1.21)	70.34 (±1.04)	85.68 (±2.55)	77.23 (±1.00)
10	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB + TUR-AG	89.16 (±1.64)	94.86 (±1.08)	91.91 (±0.77)	91.67 (±1.49)	95.10 (±1.60)	93.34 (±1.00)
11	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB + TUR-PJ	88.06 (±0.87)	94.57 (±0.91)	91.19 (±0.32)	89.81 (±1.45)	94.77 (±0.75)	92.21 (±0.48)
12	Model 4	dbmdz/bert-base-turkish-cased + CRF	TDB + TUR-AG + TUR-PJ	88.54 (±1.20)	93.84 (±1.17)	91.10 (±0.52)	90.77 (±0.62)	94.02 (±1.38)	92.36 (±0.61)
13	Model 1	dbmdz/bert-base-turkish-cased	TDB	90.69 (±0.23)	94.99 (±0.44)	92.79 (±0.20)	93.63 (±0.18)	95.22 (±0.87)	**94.42 (±0.37)**
14	Model 4	bert-base-multilingual-cased + CRF	PDTB + TDB + CDTB	85.33 (±2.80)	91.61 (±0.79)	88.33 (±1.33)	88.59 (±2.50)	92.37 (±0.98)	90.41 (±0.95)
15	Model 1	bert-base-multilingual-cased	PDTB + TDB + CDTB	85.54 (±1.03)	92.69 (±1.06)	88.96 (±0.25)	89.29 (±1.60)	93.05 (±0.62)	91.12 (±0.82)
16	Model 1	bert-base-multilingual-cased	PDTB-05% + TDB-10% + CDTB	80.57 (±1.80)	89.02 (±0.78)	84.57 (±1.01)	84.24 (±1.53)	88.17 (±1.31)	86.16 (±1.21)
17	—	baseline	TDB	—	—	—	47.64	33.22	39.14

Table 4.2: Performance of various model configurations on the TDB (Turkish) development and test sets.

It is clear from the experiments that the synthetic datasets do not lead to an increase in performance on this task. The projection data set (TUR-PJ) is particularly unrepresentative of the target task, achieving an F-measure of 41.15 with model 4 (row 8) on the test set, which is only 2 points above the baseline of selecting the most probable classification based on the frequency of DU and NDU in the respective training set (F-measure of 39.14). This seems to indicate that the projection method employed does not project useful DC annotations. On the other hand, the agreement method does produce a better dataset (TUR-AG) than the projection method, attaining an F-measure of 87.99 on the test set (model 4 on row 7). However, it does not seem to provide an additional performance increase when trained with the TDB (model 4 on row 10) with an F-measure of 93.34 compared to 94.01 (model 4 on row 1). Both methods perform more poorly than when the model is trained only on 10% of the TDB (model 4 on row 5). Section 4.2.1.2 will analyse this further.

The experiments with model 4 and the reduced TDB (rows 2 to 6) indicates that the task of Turkish DC identification does not require all that much data to have a strong performance. TDB-05% only contains 1248 training instances and 319 tokens annotated as `B-Conn` and 60 annotated as `I-Conn`, yet, achieves an F-measure of 86.04 (row 7). This may indicate that most of the DCs in the test set are common `Explicit` DCs, since the model only has to learn these to achieve a good performance on the test set.

Multilingual cross-training appears to have a negative impact on the task of Turkish DC identification (rows 14, 15, and 16). Indeed, row 14 performs 3.60% lower than the same model trained solely on the TDB (row 1). Similarly, row 15 performs 3.30% lower than its counterpart (row 13). This is believed to be because of the imbalance between the size of each of the data sets. The PDTB contains over 40k instances where as the TDB contains only 20k. To adjust for this imbalance, in row 16, each dataset contains around 2k training instances; yet, the model performs worse than with the imbalanced data. This seems to indicates that multilingual cross training with languages that are not part of the same family of languages may not transfer useful information for the task of Turkish DC identification.

4.1.3 Chinese Results

Table 4.3 shows the performance of various models on the CDTB development and test sets. In general, the Chinese models have the lowest performance in DC identification. The model that achieves the best performance is BERT (`bert-base-chinese`) with a CRF output layer (model 4 on row 1), which achieves a performance of 87.47 with a standard deviation of 0.96 on the test set and 86.39 with a standard deviation of 0.44 on the development set. The test set overall leads to stronger F-measures compared to the development set, indicating that the test set is an easier task.

The synthetic Chinese data sets again do not seem to represent the task of Chinese DC identification well, and both methods have poorer performance compared to using the CDTB only. The projection method (ZHO-PJ) achieves an F-measure of 57.37 (row 8). This is better than that of the Turkish projection dataset for Turkish DC identification (see row 8 of Table 4.2) which had a F-measure of 41.14. The baseline of choosing the most probable classification based on the frequency of DUs and NDUs in the respective training sets achieves an F-measure of 55.50 on the CDTB test set, which is only 1.87 points worse than the projection data set. It is clear that the Chinese synthetic projection data set (ZHO-PJ) does not contain valuable DC annotations. The synthetic agreement data set (ZHO-AG) fares better, achieving an F-measure of 86.31 (row 7) only 1.16 points behind the best performing Chinese model. This strong result is likely attributed to the lack of error accumulation due to not having to generate word alignments. Additionally, when using the agreement dataset in conjunction with the CDTB (row 9), the F-measure only decreases by 0.01 points, indicating that this dataset might bring interesting information.

The poor performance of training on all the CDTB (rows 1 and 13) is likely due to the small size of the data set and the particularities of Mandarin Chinese, where most DCs have two forms (discontinuous and single), both of which can be used interchangeably (see Section 2.2.2). Additionally, the task of identifying DCs in Chinese involves identifying `Explicit` and `AltLex` connectives. As the CDTB is already quite small, the results from the experiments with reducing it further (rows 2 to 6) were not all that surprising. Each reduction in size resulted in a much weaker model, excluding the CDTB-75% (row 2). CDTB-75% achieves the strongest performance on the

				CDTB Dev Set			CDTB Test Set		
Row	**Model #**	**Model**	**Training Dataset**	**Dev Precision**	**Dev Recall**	**Dev F-measure**	**Test Precision**	**Test Recall**	**Test F-measure**
1	Model 4	bert-base-chinese + CRF	CDTB	86.47 (±0.57)	86.31 (±0.39)	86.39 (±0.44)	88.43 (±1.67)	86.54 (±0.72)	**87.47 (±0.96)**
2	Model 4	bert-base-chinese + CRF	CDTB-75%	85.17 (±0.50)	87.07 (±0.73)	86.11 (±0.50)	88.33 (±1.34)	86.80 (±0.53)	87.55 (±0.88)
3	Model 4	bert-base-chinese + CRF	CDTB-50%	83.14 (±0.64)	83.82 (±1.48)	83.47 (±0.73)	84.42 (±1.17)	85.39 (±0.54)	84.89 (±0.67)
4	Model 4	bert-base-chinese + CRF	CDTB-25%	77.73 (±4.51)	79.36 (±1.61)	78.49 (±2.80)	81.16 (±2.97)	81.28 (±0.74)	81.20 (±1.62)
5	Model 4	bert-base-chinese + CRF	CDTB-10%	69.35 (±2.59)	78.79 (±1.89)	73.75 (±2.02)	73.13 (±2.89)	81.60 (±1.05)	77.10 (±1.56)
6	Model 4	bert-base-chinese + CRF	CDTB-05%	65.89 (±2.48)	73.44 (±0.73)	69.44 (±1.46)	69.48 (±2.35)	76.73 (±1.48)	72.89 (±1.09)
7	Model 4	bert-base-chinese + CRF	ZHO-AG	78.10 (±1.59)	80.32 (±1.38)	79.19 (±1.15)	87.00 (±1.75)	85.64 (±0.73)	86.31 (±1.10)
8	Model 4	bert-base-chinese + CRF	ZHO-PJ	59.95 (±3.05)	48.28 (±3.13)	53.35 (±1.12)	62.47 (±3.27)	53.27 (±4.87)	57.37 (±2.94)
9	Model 4	bert-base-chinese + CRF	ZHO-AG + ZHO-PJ	75.73 (±1.20)	78.02 (±0.75)	76.85 (±0.79)	85.48 (±0.83)	84.55 (±1.39)	85.01 (±1.07)
10	Model 4	bert-base-chinese + CRF	CDTB + ZHO-AG	84.37 (±1.34)	81.78 (±1.43)	83.05 (±1.18)	89.87 (±1.35)	85.20 (±0.57)	87.46 (±0.79)
11	Model 4	bert-base-chinese + CRF	CDTB + ZHO-PJ	80.29 (±1.42)	81.66 (±0.43)	80.97 (±0.84)	86.50 (±1.61)	83.97 (±0.88)	85.21 (±0.81)
12	Model 4	bert-base-chinese + CRF	CDTB + ZHO-AG + ZHO-PJ	81.36 (±1.00)	81.78 (±2.18)	81.57 (±1.49)	87.79 (±0.92)	84.81 (±1.93)	86.27 (±1.20)
13	Model 1	bert-base-chinese	CDTB	86.01 (±2.27)	85.92 (±1.85)	85.93 (±0.60)	87.74 (±2.12)	85.77 (±1.86)	86.71 (±1.00)
14	Model 4	bert-base-multilingual-cased + CRF	PDTB + TDB + CDTB	82.98 (±2.12)	80.70 (±1.02)	81.80 (±0.69)	83.30 (±3.67)	83.46 (±1.55)	83.33 (±1.58)
15	Model 1	bert-base-multilingual-cased	PDTB + TDB + CDTB	83.02 (±1.97)	81.72 (±1.60)	82.35 (±1.42)	84.25 (±2.30)	85.19 (±1.76)	84.71 (±1.81)
16	Model 1	bert-base-multilingual-cased	PDTB-05% + TDB-10% + CDTB	83.70 (±1.64)	81.72 (±1.76)	82.69 (±1.61)	85.05 (±1.39)	86.28 (±1.05)	85.65 (±0.93)
17	—	baseline	CDTB	—	—	—	56.90	54.17	55.50

Table 4.3: Performance of various model configurations on the CDTB (Chinese) development and test sets.

test set (87.55 F-measure) but is weaker than the model trained on all the CTDB (row 1) when measured against the development set. As indicated in Section 4.1.3, the development set is a harder task, which indicates that the model on row 1 has better generalization. Although even with only 204 training instances (CTDB 10%) the performance drops 10.37 points on the training set, this shows that BERT (`bert-base-chinese`) does have some level of knowledge about DCs.

The experiments with the multilingual cross-training (rows 14 to 16) are interesting: the CRF (model 4 on row 14) seems to have a negative impact on the performance in this setting, unlike the models trained only on the CDTB (row 1 vs row 13) and comparing the results for the multilingual model without a CRF output (model 1 on row 15) we observe this behaviour. Perhaps the CRF has difficulty modeling in a multilingual setting. Additionally, similarly to Turkish, the imbalanced data does seem to impact the model's performance for Chinese DC identification. Indeed, the model trained on the full sized PDTB, TDB, and CDTB (row 15) has an F-measure 0.94 points lower than that of the model trained on the reduced PDTB and TDB where each dataset contains 2k instances for each language (row 16). Although none of the multilingual cross trained model perform better than the model trained on the CDTB alone, this again seems to indicate that multilingual cross training for DC identification with languages that are not from the same family does not provide benefit.

4.1.4 DISRPT 2021 Results

Given the results of our experiments, we selected our best models and evaluated them with the official DISRPT 2021 data set and scorer. These models are:

- For English model 4 on row 7 in Table 4.1.
- For Turkish model 1 on row 13 in Table 4.2.
- For Chinese model 1 on row 1 in Table 4.3.

Table 4.4 shows the result of the shared task of Multilingual DC identification for DISRPT 2021 and the results of the best model presented in this thesis. When comparing the performance of our models to the other participating systems at the DISRPT-2021 shared task, our base BERT models performs as well as, if not better than the top performing model, DiscoDisco (Gessler et al., 2021) for all three languages; while being significantly simpler in terms of linguistic and computational resources. The DiscoDisco approach used a collection of handcrafted features including 3 sentence embeddings (2 trainable/fine-tuned, and 1 static), a variety of grammatical and textual features (UPOS, XPOS, universal dependency relations, head distance, sentence type, and sentence length), and also a representation of the context via neighboring sentences. On the other hand, our models (model 4) are less resource-intensive, as they consist of only a language-specific BERT + CRF and only use the current sentence as context. This seems to show that the language-specific BERT-base model contains sufficient information to accomplish this task, and feeding the model with additional information is redundant and only increases its complexity without significant performance gain.

Corpus	**TMVM**			**DiscoDisco**			**disCut**			**SegFormers**			**CLACDis (best models)**			**average**
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**F1**
PDTB	85.98	65.54	74.38	92.32	91.15	92.02	93.32	88.67	90.94	89.73	92.61	91.15	93.46	92.79	**93.12**	88.32
TDB	80.00	24.14	37.10	93.71	94.53	94.11	90.55	86.93	88.70	90.42	91.17	90.79	93.63	95.22	**94.42**	81.02
CDTB	30.00	0.96	1.86	89.19	85.95	87.52	84.43	66.03	74.10	85.05	87.50	86.26	88.43	86.54	**87.47**	67.44
macro average	65.33	30.21	37.78	91.74	90.54	91.22	89.43	80.54	84.58	88.40	90.43	89.40	91.84	91.52	91.67	78.93
micro average	79.00	38.75	49.30	92.87	92.64	92.85	91.22	86.22	88.60	89.79	91.49	90.63	92.49	93.22	93.45	82.97

Table 4.4: Comparison of our best BERT models (For English model 4 on row 7 in Table 4.1, for Turkish model 1 on row 13 in Table 4.2, and for model 1 on Chinese row 1 in Table 4.3) with the official results of DISRPT 2021 Task 2, taken from Zeldes and Liu (2021).

4.2 Analysis

In order to better understand the behaviour of our models we performed a comprehensive analysis of the outputs.

4.2.1 Per-DC Analysis

The first investigation aimed at determining if there is a correlation between the performance of the model and specific DCs being tagged. For example, if the term is always in discourse usage (DU), then the DC identification task may be considered easier than a term that is used as often in discourse usage (DU) as non discourse usage (NDU). To evaluate the Pearson product-moment correlation between the F-measure of performance specific DCs, we focused on three features of DCs:

(1) The frequency of DC in the training set in DU form.

(2) The ratio of that DC being in NDU form.

(3) The entropy of the DC.

Recall that entropy (a measure of uncertainty developped by Claude Shannon) is defined as:

$$-(ratioDU * log_2 ratioDU) + (ratioNDU * log_2 ratioNDU)$$

Entropy is a measure of ambiguity; the lower that value, the more likely the DC falls consistently into either DU or NDU. Whereas the larger values represent a DC that is ambiguous; it appears in DU form as often as NDU form.

4.2.1.1 English

The per-DC analysis for English was performed using one of the five BERT large with CRF output layer (model #7 in Table 4.1). This model was used because its performance was slightly under the average of the five models, meaning more errors can be observed. Recall from Table 4.1, that this specific model had an F-measure of 93.65 on the test set and 92.28 on the development set. We used 3 inventories of DCs:

- The PDTB list of 100 DCs. (Prasad et al., 2008)
- DimLex. (Das et al., 2018)
- DCs marked by the model but not included in the above two resources.

We use the PDTB list of 100 DCs because it contains the most common occurring DCs and we expect those DC to have the best performance. We used DimLex to observe other very common DC which should also have good performance. Lastly observing the performance of the DCs that are not in those sources are mostly `AltLex` or uncommon `Explicit` DCs.

Inventory 1: PDTB Explicit DCs

The Penn Discourse Treebank 2.0 Annotation Manual[1] contains a list of 100 `Explicit` DCs. This list of `Explicit` DCs is used to identify and extract the F-measure of those DCs in the test set, to count the number of times this particular term is used in DU or NDU form, and therefore the entropy of the term, the entropy of the DC in the training set versus the F-measure of that DC in the test set. Table 4.5 shows the 100 DCs from the PDTB. Of these 100 DCs, only 2 form a Discontinued DC (#68 and #69, *on the one hand ... on the other hand*). In the PDTB, discontinued DCs are split into their components and each unit is considered a separate DC, this is how the models at the shared task and the official evaluation handles discontinuous DCs. Of the 100 PDTB DCs, 66 are found in the DISRPT 2021 test set. Table 4.6 shows the F-measure of model 4 for each DC, along with their frequency as DU and NDU in the training set, as well as their entropy. To compare how well the DCs in each lexicon perform, we use 3 different aggregations of F-measures:

- The average of the F-measures.
- The weighted average of the F-measues where the weight is based on the frequency of the DC in the test set in DU form.
- The calculated F-measure based on its true positive, false negative, and false positive rates. We sum each of these values, then calculate the precision, recall and finally calculate the F-measure of each DC.

As Table 4.6 shows, the average F-measure for this list is 88.00, whereas the weighted average is 96.23 and the calculated F-measure is 96.50. This shows that the model does not have difficulty with `Explicit` DCs. However, 6 DCs appear a total of 10 times as DU in the training set that have an F-measure of 0.00. These are: *in the end* (see Nb 61: 0 DU, 1 false positive), *specifically* (see Nb 62: 2 DU, 2 false negatives), *as well* (see Nb 63: 1 DU, 1 false negative), *further* (see Nb 64: 1 DU, 1 false negative), *for* (see Nb 65: 6 DU, 6 false negative), *in other words* (see Nb 66: 0 DU, 1 false positive). To examine if the F-measure is correlated with the specific DC, we computed the Pearson product-moment correlation coefficient between the F-measure and

(1) The frequency of DC in the training set in DU form (see column #7) - correlation of 0.099

[1] https://www.seas.upenn.edu/~pdtb/PDTBAPI/pdtb-annotation-manual.pdf

(2) The ratio of that DC being in NDU form (see column #9) - correlation of -0.351

(3) The entropy of the DC (see column #10) - correlation of 0.057

These results indicate that the F-measure does not have a strong dependency on these any of these features and the performance does not seem to be dependent on the specific DC being tagged.

1 accordingly	26 conversely	51 indeed	76 rather
2 additionally	27 earlier	52 insofar as	77 regardless
3 after	28 either	53 instead	78 separately
4 afterward	29 or	54 later	79 similarly
5 also	30 else	55 lest	80 simultaneously
6 alternatively	31 except	56 likewise	81 since
7 although	32 finally	57 meantime	82 so
8 and	33 for	58 meanwhile	83 so that
9 as	34 for example	59 moreover	84 specifically
10 as a result	35 for instance	60 much as	85 still
11 as an alternative	36 further	61 neither	86 then
12 as if	37 furthermore	62 nevertheless	87 thereafter
13 as long as	38 hence	63 next	88 thereby
14 as soon as	39 however	64 nonetheless	89 therefore
15 as though	40 if	65 nor	90 though
16 as well	41 if and when	66 now that	91 thus
17 because	42 in addition	67 on the contrary	92 till
18 before	43 in contrast	68 on the one hand	93 ultimately
19 before and after	44 in fact	69 on the other hand	94 unless
20 besides	45 in other words	70 once	95 until
21 but	46 in particular	71 or	96 when
22 by comparison	47 in short	72 otherwise	97 when and if
23 by contrast	48 in sum	73 overall	98 whereas
24 by then	49 in the end	74 plus	99 while
25 consequently	50 in turn	75 previously	100 yet

Table 4.5: 100 Explicit DC from the PDTB 2.0 annotation manual.

Inventory 2: DimLex

A similar analysis was done using the English Discourse Marker Lexicon v.1.0 (DimLex) (Das et al., 2018) which includes the 100 explicit DCs from the PDTB and an additional 43 from the RST Signalling Corpus (Das and Taboada, 2018). Table 4.7 shows the DCs that are exclusive to this lexicon. These includes Explicit and phrasal expressions (small subset of AltLex). Out of the 43 DCs in this lexicon and not in the 100 Explicit DCs from the PDTB, only 14 are

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy
1	accordingly	1.00	1.00	1.00	3	2	2	0.50	1.00
2	as a result	1.00	1.00	1.00	4	73	73	0.50	1.00
3	as though	1.00	1.00	1.00	1	5	5	0.50	1.00
4	besides	1.00	1.00	1.00	2	18	17	0.51	1.00
5	otherwise	1.00	1.00	1.00	1	23	20	0.53	1.00
6	after	1.00	1.00	1.00	23	667	548	0.55	0.99
7	nor	1.00	1.00	1.00	4	47	32	0.59	0.97
8	as soon as	1.00	1.00	1.00	2	11	17	0.39	0.97
9	finally	1.00	1.00	1.00	1	48	30	0.62	0.96
10	simultaneously	1.00	1.00	1.00	1	8	5	0.62	0.96
11	hence	1.00	1.00	1.00	1	7	4	0.64	0.95
12	later	1.00	1.00	1.00	2	156	86	0.64	0.94
13	yet	1.00	1.00	1.00	2	186	94	0.66	0.92
14	by then	1.00	1.00	1.00	1	3	6	0.33	0.92
15	rather	1.00	1.00	1.00	1	129	54	0.70	0.88
16	as if	1.00	1.00	1.00	1	6	15	0.29	0.86
17	instead	1.00	1.00	1.00	2	51	146	0.26	0.83
18	separately	1.00	1.00	1.00	3	18	68	0.21	0.74
19	now that	1.00	1.00	1.00	1	5	22	0.19	0.69
20	by comparison	1.00	1.00	1.00	1	2	9	0.18	0.68
21	in addition	1.00	1.00	1.00	11	39	176	0.18	0.68
22	for example	1.00	1.00	1.00	8	20	184	0.10	0.46
23	in fact	1.00	1.00	1.00	3	7	80	0.08	0.40
24	therefore	1.00	1.00	1.00	1	2	23	0.08	0.40
25	nonetheless	1.00	1.00	1.00	2	2	25	0.07	0.38
26	much as	1.00	1.00	1.00	1	150	9	0.94	0.31
27	for instance	1.00	1.00	1.00	13	5	84	0.06	0.31
28	so that	1.00	1.00	1.00	1	1	30	0.03	0.21
29	nevertheless	1.00	1.00	1.00	7	1	36	0.03	0.18
30	unless	1.00	1.00	1.00	1	2	94	0.02	0.15
31	meanwhile	1.00	1.00	1.00	14	2	173	0.01	0.09
32	moreover	1.00	1.00	1.00	3	1	95	0.01	0.08
33	although	1.00	1.00	1.00	16	1	302	0.00	0.03
34	additionally	1.00	1.00	1.00	1	0	6	0.00	0.00
35	by contrast	1.00	1.00	1.00	2	0	26	0.00	0.00
36	consequently	1.00	1.00	1.00	1	0	9	0.00	0.00
37	on the other hand	1.00	1.00	1.00	3	0	35	0.00	0.00
38	whereas	1.00	1.00	1.00	1	0	4	0.00	0.00
39	when	1.00	0.98	0.99	50	363	998	0.27	0.84
40	also	0.99	0.99	0.99	76	108	1615	0.06	0.34
41	and	0.98	0.99	0.98	282	12648	5829	0.68	0.90
42	but	1.00	0.96	0.98	190	362	3402	0.10	0.46
43	while	1.00	0.95	0.97	37	40	742	0.05	0.29
44	however	1.00	0.94	0.97	36	12	432	0.03	0.18
45	or	0.93	1.00	0.97	14	2508	371	0.87	0.55
46	because	1.00	0.93	0.96	44	427	792	0.35	0.93
47	if	0.94	0.98	0.96	52	136	1216	0.10	0.47
48	though	0.92	1.00	0.96	11	25	312	0.07	0.38
49	as	1.00	0.89	0.94	46	4110	884	0.82	0.67
50	before	0.94	0.94	0.94	17	336	300	0.53	1.00
51	thus	0.91	0.91	0.91	11	8	98	0.08	0.39
52	so	0.93	0.88	0.90	16	562	321	0.64	0.95
53	until	1.00	0.80	0.89	5	191	156	0.55	0.99
54	then	0.83	0.94	0.88	16	98	368	0.21	0.74
55	since	0.82	0.90	0.86	10	436	206	0.68	0.91
56	earlier	1.00	0.75	0.86	4	645	12	0.98	0.13
57	still	0.83	0.83	0.83	12	491	175	0.74	0.83
58	once	1.00	0.67	0.80	3	144	79	0.65	0.94
59	indeed	0.67	1.00	0.80	2	22	97	0.18	0.69
60	previously	0.50	1.00	0.67	3	110	49	0.69	0.89
61	in the end	0.00	0.00	0.00	0	10	11	0.48	1.00
62	specifically	0.00	0.00	0.00	2	16	9	0.64	0.94
63	as well	0.00	0.00	0.00	1	211	24	0.90	0.48
64	further	0.00	0.00	0.00	1	272	13	0.95	0.27
65	for	0.00	0.00	0.00	6	9093	339	0.96	0.22
66	in other words	0.00	0.00	0.00	0	0	16	0.00	0.00
Average F-measure				0.88					
Weighted Average F-measure				0.96					
Calculated F-measure				0.97					

Table 4.6: Performance of model on row #7 in Table 4.1 for the 66 PDTB `Explicit` DCs found in the test set with their frequency and entropy per DC.

found in the DISRPT 2021 test set. Table 4.8 shows the performance of these DCs and how often they are found in the training set in DU or NDU form as well as their entropy. As the table shows, the average F-measure is 70.10, whereas the weighted average is 86.12 and the calculated the F-measure is 85.71. Most of these DCs do appear in the training set in NDU form more often than in DU, making learning when to annotate these DCs ambiguous. However, when we calculate the correlation between the F-measure and various features of the DC, we can see that this does not seem to be the case. The correlation between the frequency of DU in the training set and the F-measure is 0.2651, between the NDU ratio and the F-measure is -0.5305 and between the entropy and the F-measure 0.0025. The strongest correlation is with the NDU ratio in the training set, indicating that the F-measure is slightly affected by how many times a DC is in NDU form over all of its appearances.

1 after all	12 despite	23 given that	34 not but
2 after that	13 essentially	24 in addition to	35 not only
3 afterwards	14 even if	25 in any case	36 particularly
4 anyway	15 even so	26 in any event	37 quite the contrary
5 as a result of	16 even though	27 in case	38 rather than
6 aside from	17 eventually	28 in essence	39 upon
7 at that point	18 everytime	29 in response to	40 whatever
8 at the same time	19 except that	30 in spite of	41 whenever
9 at the time	20 for one	31 in this way	42 with
10 because of	21 for one thing	32 instead of	43 without
11 by the way	22 given	33 irrespective of	

Table 4.7: DimLex DCs that are not part of the 100 DCs from PDTB 2.0 annotation manual.

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
1	whenever	1.00	1.00	1.00	1	7	7	0.50	1.00
2	not only	1.00	1.00	1.00	1	35	37	0.51	1.00
3	instead of	1.00	1.00	1.00	1	41	50	0.55	0.99
4	after that	1.00	1.00	1.00	1	4	8	0.67	0.92
5	at the same time	1.00	1.00	1.00	6	58	9	0.13	0.57
6	given	1.00	1.00	1.00	1	10	149	0.94	0.34
7	even though	1.00	1.00	1.00	5	88	0	0.00	0.00
8	even if	1.00	1.00	1.00	1	83	0	0.00	0.00
9	without	0.75	1.00	0.86	3	90	231	0.72	0.86
10	with	0.75	1.00	0.86	6	304	4584	0.94	0.34
11	rather than	0.25	1.00	0.40	1	38	96	0.72	0.86
12	for one	0.00	0.00	0.00	1	10	45	0.82	0.68
13	at the time	0.00	0.00	0.00	1	7	46	0.87	0.56
14	eventually	0.00	0.00	0.00	0	7	71	0.91	0.44
Average F-measure				0.70					
Weighted Average F-measure				0.86					
Calculated F-measure				0.86					

Table 4.8: Performance of model on row #7 in Table 4.1 for DimLex only DCs found in the test set with their frequency and entropy per DC.

Inventory 3: DCs In Neither Lexicon

The model has identified 126 DCs in the test set that are not found in neither of the lexicons (PDTB or DimLex). Tables 4.9 and 4.10 show the performances of each of these DCs. The average F-measure is a low 38.24, the weighted average is 61.55, and the calculated F-measure is 59.40. Most of these DCs do not appear in the training set in DU form, or only have few occurrences where they are in DU form, on average having 15.83 occurrences of DU compared to an average of 1118.54 occurrences of NDU. However, the correlation between the F-measure and the frequency of DU in the training set is 0.0382, the ratio of NDU is -0.0227 and entropy 0.4157, shows weak or no correlation.

Figures 4.1, 4.2, and 4.3 summarises the correlation of the performance with each DCin each DC inventory. Figure 4.1 shows the frequency of DU for each DC in the training set versus the F-measure of that DC in the test set. The reasoning is that the more positive instances (annotated as DU) a DC has, the more confident the model should be about that DC being in discourse usage. Although this ignores the question of ambiguity, focusing only on the frequency of DU.

Figure 4.2 shows the ratio of NDU for the DC in the training set versus the F-measure of that DC in the test set. We expected a strong negative correlation because the larger the ratio of NDU for a DC is, the more likely that the DC is used in NDU. However, there seems to be only a small negative correlation. This measure takes into account the ambiguity of a DC.

Figure 4.3 shows the entropy vs the F-measure of the DCs. As entropy is a measure of ambiguity, we expected DCs with a lower entropy (less ambiguous) to have a better performance. While this is not always the case, there does seem to be a small cluster of points at the top left corner of the graph.

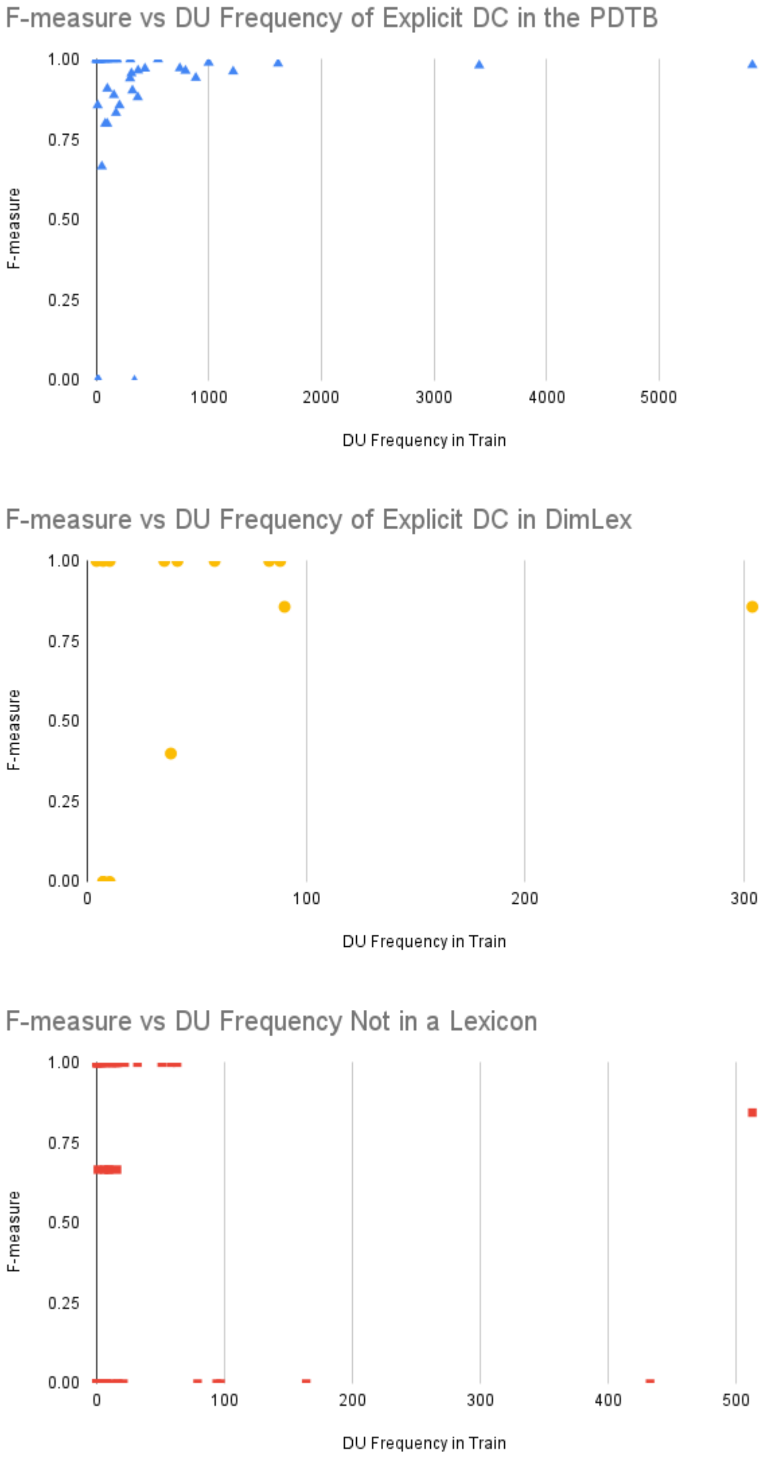
Note that Figures 4.1, 4.2, and 4.3, may not indicate a correlation between the features of these DCs with the F-measure, but they do seem to indicate that DCs that are not in any lexicon are much more difficult for the model to identify. This is likely because the 100 PDTB and DimLex connectives are `Explicit` connectives and common phrasal expressions; whereas the other are a combination of less common `AltLex` and errors done by the model.

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
1	that means	1.00	1.00	1.00	2	15	15	0.50	1.00
2	since then	1.00	1.00	1.00	2	17	16	0.48	1.00
3	partly because	1.00	1.00	1.00	2	16	21	0.57	0.99
4	not because	1.00	1.00	1.00	1	3	2	0.40	0.97
5	the result is	1.00	1.00	1.00	1	4	6	0.60	0.97
6	in part because	1.00	1.00	1.00	2	10	6	0.38	0.95
7	leaving	1.00	1.00	1.00	2	22	37	0.63	0.95
8	causing	1.00	1.00	1.00	1	14	25	0.64	0.94
9	especially after	1.00	1.00	1.00	1	1	2	0.67	0.92
10	filling	1.00	1.00	1.00	1	5	10	0.67	0.92
11	only to	1.00	1.00	1.00	1	16	32	0.67	0.92
12	even after	1.00	1.00	1.00	1	7	3	0.30	0.88
13	just when	1.00	1.00	1.00	1	5	2	0.29	0.86
14	indicating	1.00	1.00	1.00	1	8	21	0.72	0.85
15	reflecting	1.00	1.00	1.00	1	58	20	0.26	0.82
16	so are	1.00	1.00	1.00	1	3	1	0.25	0.81
17	resulting in	1.00	1.00	1.00	2	8	2	0.20	0.72
18	in reaction	1.00	1.00	1.00	1	1	6	0.86	0.59
19	making	1.00	1.00	1.00	2	32	231	0.88	0.53
20	increasing	1.00	1.00	1.00	1	12	97	0.89	0.50
21	suggesting	1.00	1.00	1.00	1	2	17	0.89	0.49
22	followed by	1.00	1.00	1.00	1	4	35	0.90	0.48
23	in order	1.00	1.00	1.00	2	51	5	0.09	0.43
24	even as	1.00	1.00	1.00	1	12	1	0.08	0.39
25	even with	1.00	1.00	1.00	1	1	12	0.92	0.39
26	provided	1.00	1.00	1.00	1	4	80	0.95	0.28
27	not	1.00	1.00	1.00	1	63	1480	0.96	0.25
28	producing	1.00	1.00	1.00	1	2	53	0.96	0.23
29	trying	1.00	1.00	1.00	2	6	204	0.97	0.19
30	as did	1.00	1.00	1.00	1	9	0	0.00	0.00
31	as do	1.00	1.00	1.00	2	1	0	0.00	0.00
32	especially when	1.00	1.00	1.00	1	3	0	0.00	0.00
33	exacerbating	1.00	1.00	1.00	1	1	0	0.00	0.00
34	further fueling	1.00	1.00	1.00	1	0	0	0.00	0.00
35	greatly expanding	1.00	1.00	1.00	1	0	0	0.00	0.00
36	just a month after	1.00	1.00	1.00	1	0	0	0.00	0.00
37	one reason is	1.00	1.00	1.00	2	0	0	0.00	0.00
38	only if	1.00	1.00	1.00	1	12	0	0.00	0.00
39	particularly after	1.00	1.00	1.00	1	0	0	0.00	0.00
40	perhaps because	1.00	1.00	1.00	1	1	0	0.00	0.00
41	presumably because	1.00	1.00	1.00	1	0	0	0.00	0.00
42	propelled by	1.00	1.00	1.00	1	0	1	1.00	0.00
43	that 's because	1.00	1.00	1.00	1	15	0	0.00	0.00
44	threatening	1.00	1.00	1.00	1	0	19	1.00	0.00
45	by	0.73	1.00	0.84	19	513	4630	0.90	0.47
46	assuming	0.50	1.00	0.67	1	16	16	0.50	1.00
47	largely because	1.00	0.50	0.67	2	9	14	0.61	0.97
48	bringing	1.00	0.50	0.67	2	10	29	0.74	0.82
49	reducing	0.50	1.00	0.67	1	6	35	0.85	0.60
50	at this point	0.50	1.00	0.67	1	1	11	0.92	0.41
51	so new	0.00	0.00	0.00	1	1	1	0.50	1.00
52	typical is	0.00	0.00	0.00	0	1	1	0.50	1.00
53	aided by	0.00	0.00	0.00	0	6	5	0.45	0.99
54	that 's why	0.00	0.00	0.00	0	4	3	0.43	0.99
55	what 's more	0.00	0.00	0.00	0	11	7	0.39	0.96
56	at that time	0.00	0.00	0.00	0	3	8	0.73	0.85
57	will result in	0.00	0.00	0.00	1	3	8	0.73	0.85
58	as was	0.00	0.00	0.00	1	1	4	0.80	0.72
59	forcing	0.00	0.00	0.00	1	4	17	0.81	0.70
60	partly	0.00	0.00	0.00	0	17	80	0.82	0.67
61	especially	0.00	0.00	0.00	1	21	126	0.86	0.59
62	but because	0.00	0.00	0.00	0	7	1	0.13	0.54
63	partly because of	0.00	0.00	0.00	1	2	21	0.91	0.43
64	too	0.00	0.00	0.00	2	16	347	0.96	0.26
65	whether	0.00	0.00	0.00	0	7	299	0.98	0.16

Table 4.9: Performance of model on row #7 in Table 4.1 for the DCs not in the the PDTB and DimLex found in the test set with their frequency and entropy per DC.

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
66	in	0.00	0.00	0.00	2	433	18498	0.98	0.16
67	that	0.00	0.00	0.00	0	164	9222	0.98	0.13
68	both	0.00	0.00	0.00	0	8	533	0.99	0.11
69	such as	0.00	0.00	0.00	0	2	376	0.99	0.05
70	to	0.00	0.00	0.00	0	97	24888	1.00	0.04
71	a	0.00	0.00	0.00	0	79	22293	1.00	0.03
72	will	0.00	0.00	0.00	1	8	3414	1.00	0.02
73	the	0.00	0.00	0.00	0	94	53321	1.00	0.02
74	a major reason is	0.00	0.00	0.00	1	0	1	1.00	0.00
75	another is	0.00	0.00	0.00	0	0	2	1.00	0.00
76	another is that	0.00	0.00	0.00	1	0	0	0.00	0.00
77	as evidenced by	0.00	0.00	0.00	1	0	1	1.00	0.00
78	as has	0.00	0.00	0.00	1	2	0	0.00	0.00
79	as has been	0.00	0.00	0.00	0	0	0	0.00	0.00
80	as was the	0.00	0.00	0.00	0	0	1	1.00	0.00
81	but also because	0.00	0.00	0.00	0	0	0	0.00	0.00
82	chalking up	0.00	0.00	0.00	0	0	1	1.00	0.00
83	coming as	0.00	0.00	0.00	0	1	0	0.00	0.00
84	drawing	0.00	0.00	0.00	1	0	16	1.00	0.00
85	effective	0.00	0.00	0.00	0	0	93	1.00	0.00
86	examples are	0.00	0.00	0.00	1	0	0	0.00	0.00
87	further pressuring	0.00	0.00	0.00	0	0	0	0.00	0.00
88	further squeezing	0.00	0.00	0.00	1	0	0	0.00	0.00
89	further supporting	0.00	0.00	0.00	1	0	0	0.00	0.00
90	in a similar vein	0.00	0.00	0.00	1	0	0	0.00	0.00
91	indicated that	0.00	0.00	0.00	1	0	30	1.00	0.00
92	largely reflecting	0.00	0.00	0.00	0	0	1	1.00	0.00
93	on the bottom line	0.00	0.00	0.00	0	0	1	1.00	0.00
94	potentially	0.00	0.00	0.00	0	0	27	1.00	0.00
95	potentially exempting	0.00	0.00	0.00	1	0	0	0.00	0.00
96	should the courts uphold the validity	0.00	0.00	0.00	0	0	0	0.00	0.00
97	should the courts uphold the validity of this type of defense	0.00	0.00	0.00	1	0	0	0.00	0.00
98	so , too	0.00	0.00	0.00	1	0	0	0.00	0.00
99	so as	0.00	0.00	0.00	0	4	0	0.00	0.00
100	so oriented as	0.00	0.00	0.00	1	0	0	0.00	0.00
101	squeezing	0.00	0.00	0.00	0	0	3	1.00	0.00
102	such a hard time counting all the planes in their fleets	0.00	0.00	0.00	1	0	0	0.00	0.00
103	supporting	0.00	0.00	0.00	0	0	27	1.00	0.00
104	that 's when	0.00	0.00	0.00	0	0	1	1.00	0.00
105	that action	0.00	0.00	0.00	1	0	0	0.00	0.00
106	that change will obviously impact	0.00	0.00	0.00	1	0	0	0.00	0.00
107	that follows	0.00	0.00	0.00	0	0	2	1.00	0.00
108	that ranks	0.00	0.00	0.00	0	0	0	0.00	0.00
109	that rise came on top of	0.00	0.00	0.00	1	0	0	0.00	0.00
110	that was modestly higher than	0.00	0.00	0.00	1	0	0	0.00	0.00
111	the announcement caused	0.00	0.00	0.00	1	0	0	0.00	0.00
112	the cuts are necessary	0.00	0.00	0.00	1	0	0	0.00	0.00
113	the decision	0.00	0.00	0.00	1	0	44	1.00	0.00
114	the delay resulted from	0.00	0.00	0.00	1	0	0	0.00	0.00
115	the main reason remains	0.00	0.00	0.00	1	0	0	0.00	0.00
116	three days later	0.00	0.00	0.00	1	0	3	1.00	0.00
117	thus forcing	0.00	0.00	0.00	0	0	0	0.00	0.00
118	to make its point	0.00	0.00	0.00	1	0	0	0.00	0.00
119	too late	0.00	0.00	0.00	1	0	9	1.00	0.00
120	toward that end	0.00	0.00	0.00	0	0	0	0.00	0.00
121	trapping	0.00	0.00	0.00	0	0	0	0.00	0.00
122	was one reason for the downgrade	0.00	0.00	0.00	1	0	0	0.00	0.00
123	were cited	0.00	0.00	0.00	1	0	3	1.00	0.00
124	what has changed is that	0.00	0.00	0.00	1	0	0	0.00	0.00
125	which can result in	0.00	0.00	0.00	1	0	0	0.00	0.00
126		0.00	0.00	0.00	0	0	0	0.00	0.00
Average F-measure				0.38					
Weighted Average F-measure				0.62					
Calculated F-measure				0.59					

Table 4.10: Performance of model on row #7 in Table 4.1 for the DCs not in the PDTB and DimLex found in the test set with their frequency and entropy per DC.


F-measure vs DU Frequency of Explicit DC in the PDTB

F-measure vs DU Frequency of Explicit DC in DimLex

F-measure vs DU Frequency Not in a Lexicon

Figure 4.1: Frequency of English DUs in training set of each DC vs F-measure for PDTB test set.

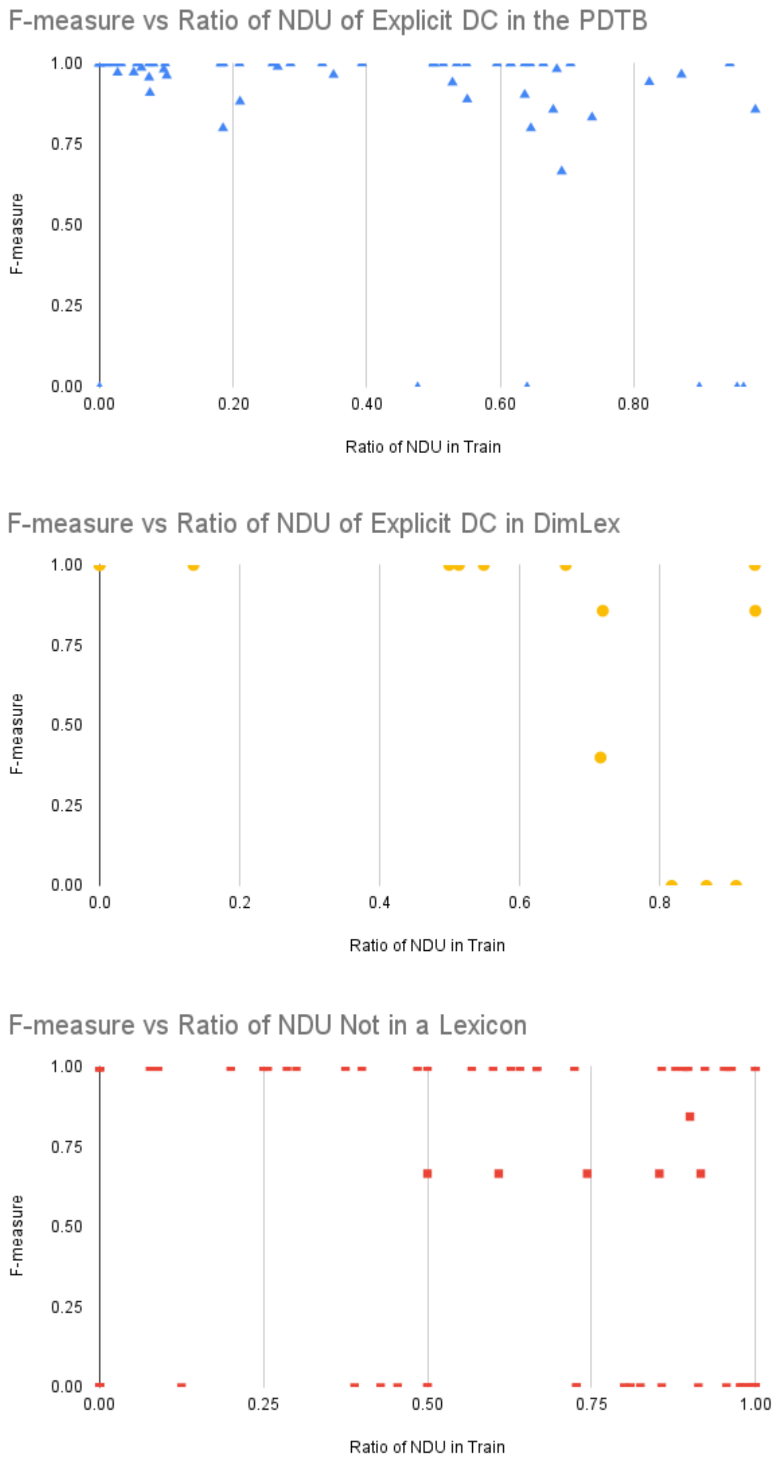
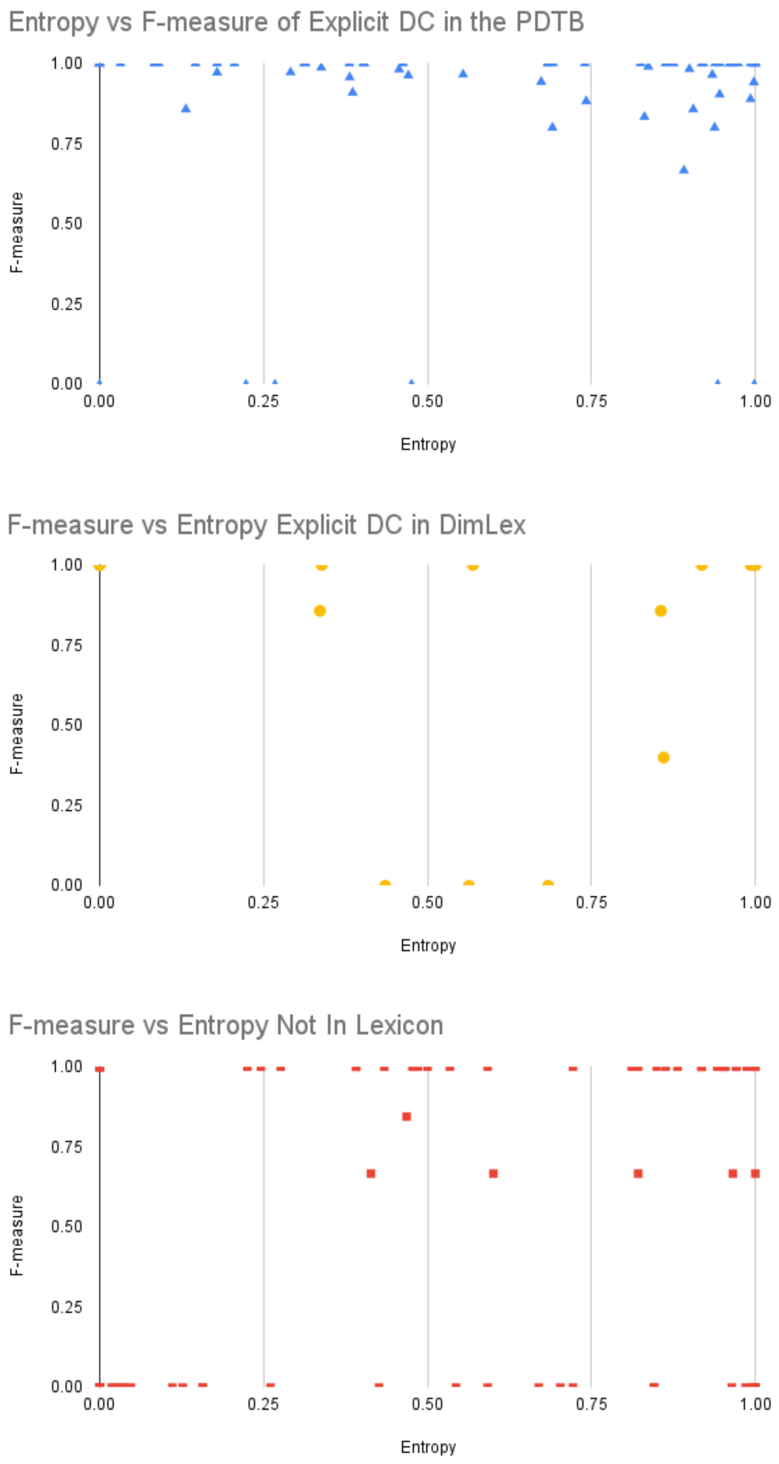


Figure 4.2: Ratio of English NDUs of each DC vs F-measure for PDTB test set.



Entropy vs F-measure of Explicit DC in the PDTB

F-measure vs Entropy Explicit DC in DimLex

F-measure vs Entropy Not In Lexicon

Figure 4.3: Entropy of each English DC vs F-measure for PDTB test set.

4.2.1.2 Turkish

We also analysed the correlation between features of DCs and the model performance for Turkish. To do this, we used model #13 from Table 4.2 and used the Turkish connective Lexicon (TCL) (Zeyrek and Başıbüyük, 2019) as DC inventory.

Turkish Connective Lexicon

The Turkish Connective Lexicon (TCL) (Zeyrek and Başıbüyük, 2019), contains 226 discourse connectives, from the `Explicit` DCs of the TDB 1.0/1.1, and the TED-MDB (see Section 2.2.4.1). Out of these, 44 are marked by our Turkish model and found in the TCL while; 59 are marked but not in the TCL. Table 4.11 shows the performance of each DC in the lexicon. The average F-measure is 90.50, the weighted average is 95.66 and the calculated F-measure is 95.74. Again, this indicates that the model can easily identify `Explicit` DCs. The correlation between the F-measure and the frequency of DUs in the train set is 0.1197, between the F-measure and the ratio of NDUs in the training set is -0.3516, while the correlation between the F-measure and entropy is 0.3987. Again, these features do not seem to correlate with the performance of the TCL DC in the test set. Figures 4.4, 4.5, 4.6, show this visually.

Inventory 2: DCs Not Found in the TCL

Similarly, we can observe the performance of the DCs that are not in the TCL, of which there are 59 marked by the model (see Table 4.12). For these, the average F-measure is 65.90, the weighted average is 80.84, and the calculated F-measure is 73.12. The correlation between the frequency of DUs and the F-measure us 0.0986, the ratio of NDU and the F-measure is -0.2603 and the entropy and the F-measure is 0.1083. Again, these features do not seem to correlate with the performance of the model on the test set.

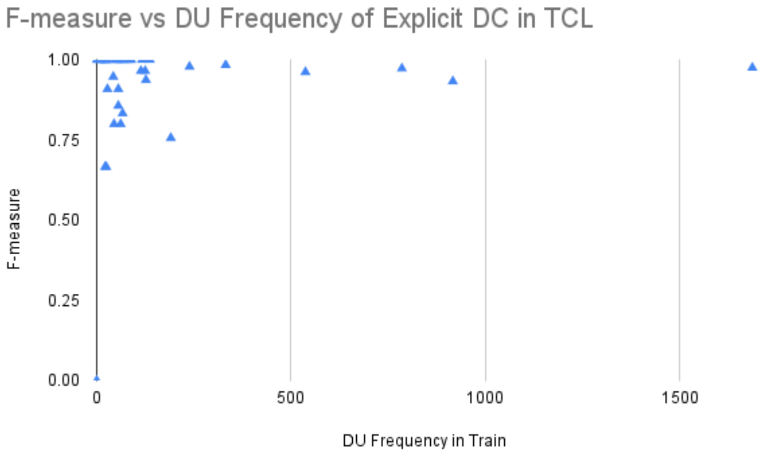
Figure 4.4 shows the frequency of DUs for each DC in the training set versus the F-measure of that DC in the test set. Again, if a DC occurs often as DU in the training set, then we expect the model to be capable of identifying it as a DU or NDU. However the graph does not show this type

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
1	hemen önce	1.00	1.00	1.00	1	1	1	0.50	1.00
2	rağmen	1.00	1.00	1.00	12	58	46	0.44	0.99
3	tersine	1.00	1.00	1.00	1	9	12	0.57	0.99
4	mesela	1.00	1.00	1.00	1	12	8	0.40	0.97
5	için de	1.00	1.00	1.00	2	25	41	0.62	0.96
6	hem	1.00	1.00	1.00	8	80	156	0.66	0.92
7	zaman	1.00	1.00	1.00	9	137	360	0.72	0.85
8	dolayısıyla	1.00	1.00	1.00	6	54	18	0.25	0.81
9	sonra da	1.00	1.00	1.00	13	61	18	0.23	0.77
10	amacıyla	1.00	1.00	1.00	2	54	13	0.19	0.71
11	halde	1.00	1.00	1.00	3	49	11	0.18	0.69
12	beraber	1.00	1.00	1.00	1	5	24	0.83	0.66
13	ayrıca	1.00	1.00	1.00	6	86	14	0.14	0.58
14	sanki	1.00	1.00	1.00	2	22	160	0.88	0.53
15	fakat	1.00	1.00	1.00	5	71	9	0.11	0.51
16	birlikte	1.00	1.00	1.00	3	29	252	0.90	0.48
17	böylece	1.00	1.00	1.00	10	70	8	0.10	0.48
18	ne var ki	1.00	1.00	1.00	2	27	3	0.10	0.47
19	ne	1.00	1.00	1.00	4	89	957	0.91	0.42
20	halbuki	1.00	1.00	1.00	1	16	1	0.06	0.32
21	gene de	1.00	1.00	1.00	3	22	1	0.04	0.26
22	bir yandan da	1.00	1.00	1.00	2	25	1	0.04	0.24
23	oysa	1.00	1.00	1.00	5	117	3	0.03	0.17
24	ancak	1.00	0.97	0.98	32	332	81	0.20	0.71
25	çünkü	1.00	0.96	0.98	24	239	5	0.02	0.14
26	ve	0.97	0.98	0.98	223	1686	4337	0.72	0.86
27	ama	0.96	0.99	0.97	110	785	85	0.10	0.46
28	ya da	0.93	1.00	0.97	14	114	214	0.65	0.93
29	önce	1.00	0.93	0.97	15	125	463	0.79	0.75
30	sonra	0.93	1.00	0.96	76	537	481	0.47	1.00
31	daha sonra	1.00	0.90	0.95	10	43	24	0.36	0.94
32	kadar	0.94	0.94	0.94	16	127	706	0.85	0.62
33	için	0.88	1.00	0.93	84	916	850	0.48	1.00
34	karşın	0.83	1.00	0.91	5	56	34	0.38	0.96
35	hem de	1.00	0.83	0.91	6	28	63	0.69	0.89
36	yine de	1.00	0.75	0.86	4	56	2	0.03	0.22
37	ardından	0.71	1.00	0.83	5	67	113	0.63	0.95
38	aslında	0.67	1.00	0.80	6	62	38	0.38	0.96
39	örneğin	0.67	1.00	0.80	4	45	13	0.22	0.77
40	gibi	0.82	0.70	0.76	20	191	1092	0.85	0.61
41	bir süre sonra	0.67	0.67	0.67	3	25	14	0.36	0.94
42	öte yandan	0.50	1.00	0.67	1	22	1	0.04	0.26
43	özellikle	0.00	0.00	0.00	1	1	125	0.99	0.07
44	artık	0.00	0.00	0.00	1	0	310	1.00	0.00
Average F-measure				0.91					
Weighted Average F-measure				0.96					
Calculated F-measure				0.96					

Table 4.11: Performance of model in row #13 in Table 4.2 for Turkish TCL DCs found in the test set with their frequency and entropy per DC.

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
1	için”	1.00	1.00	1.00	2	1	1	0.50	1.00
2	bundan sonra	1.00	1.00	1.00	1	12	14	0.54	1.00
3	sonuçta	1.00	1.00	1.00	1	9	7	0.44	0.99
4	aksine	1.00	1.00	1.00	2	8	6	0.43	0.99
5	ilkin	1.00	1.00	1.00	1	3	2	0.40	0.97
6	o zaman	1.00	1.00	1.00	3	76	42	0.36	0.94
7	ne de	1.00	1.00	1.00	2	24	45	0.65	0.93
8	dolayı	1.00	1.00	1.00	2	17	34	0.67	0.92
9	zaman da	1.00	1.00	1.00	1	3	6	0.67	0.92
10	ya	1.00	1.00	1.00	1	152	347	0.70	0.89
11	için mi	1.00	1.00	1.00	1	8	2	0.20	0.72
12	yıllar sonra	1.00	1.00	1.00	1	4	16	0.80	0.72
13	ondan sonra	1.00	1.00	1.00	1	7	1	0.13	0.54
14	bu nedenle de	1.00	1.00	1.00	1	8	1	0.11	0.50
15	bunun için	1.00	1.00	1.00	2	25	3	0.11	0.49
16	sayesinde	1.00	1.00	1.00	1	2	23	0.92	0.40
17	bir yandan	1.00	1.00	1.00	1	46	3	0.06	0.33
18	buna karşılık	1.00	1.00	1.00	2	17	1	0.06	0.31
19	bu yüzden	1.00	1.00	1.00	2	50	2	0.04	0.24
20	bu nedenle	1.00	1.00	1.00	7	84	2	0.02	0.16
21	aksi halde	1.00	1.00	1.00	1	5	0	0.00	0.00
22	belki de o nedenle	1.00	1.00	1.00	2	0	0	0.00	0.00
23	bir yıl sonra da	1.00	1.00	1.00	1	0	0	0.00	0.00
24	birkaç yıl sonra da	1.00	1.00	1.00	1	0	0	0.00	0.00
25	bu yüzden de	1.00	1.00	1.00	1	3	0	0.00	0.00
26	bunun neticesinde	1.00	1.00	1.00	1	0	0	0.00	0.00
27	daha sonra da	1.00	1.00	1.00	1	7	0	0.00	0.00
28	daha sonra ise	1.00	1.00	1.00	1	0	0	0.00	0.00
29	iki gün sonra	1.00	1.00	1.00	1	0	1	1.00	0.00
30	iki yıl sonra	1.00	1.00	1.00	1	0	1	1.00	0.00
31	ilk önce	1.00	1.00	1.00	1	1	0	0.00	0.00
32	işte o zaman	1.00	1.00	1.00	3	2	0	0.00	0.00
33	işte o zaman da	1.00	1.00	1.00	1	0	0	0.00	0.00
34	kadar da	1.00	1.00	1.00	1	0	12	1.00	0.00
35	o halde	1.00	1.00	1.00	1	3	0	0.00	0.00
36	tam aksine	1.00	1.00	1.00	1	0	0	0.00	0.00
37	aynı zamanda	0.90	0.82	0.86	11	23	22	0.49	1.00
38	onun için	0.75	1.00	0.86	3	9	13	0.59	0.98
39	buna karşın	1.00	0.50	0.67	2	13	0	0.00	0.00
40	daha önce	1.00	0.33	0.50	6	15	48	0.76	0.79
41	ardından da	0.00	0.00	0.00	1	5	5	0.50	1.00
42	her şeye rağmen	0.00	0.00	0.00	1	1	1	0.50	1.00
43	öte	0.00	0.00	0.00	1	23	33	0.59	0.98
44	buna	0.00	0.00	0.00	0	42	114	0.73	0.84
45	az sonra	0.00	0.00	0.00	1	2	6	0.75	0.81
46	sonuç	0.00	0.00	0.00	0	4	39	0.91	0.45
47	aynı	0.00	0.00	0.00	0	23	322	0.93	0.35
48	o	0.00	0.00	0.00	0	88	1289	0.94	0.34
49	11	0.00	0.00	0.00	0	0	35	1.00	0.00
50	”veya	0.00	0.00	0.00	1	0	0	0.00	0.00
51	15 yıl sonra	0.00	0.00	0.00	1	0	0	0.00	0.00
52	az önce	0.00	0.00	0.00	1	0	16	1.00	0.00
53	buna ek olarak	0.00	0.00	0.00	1	0	0	0.00	0.00
54	eğer	0.00	0.00	0.00	1	0	82	1.00	0.00
55	itibaren	0.00	0.00	0.00	0	0	65	1.00	0.00
56	o yüzden	0.00	0.00	0.00	1	7	0	0.00	0.00
57	sonuç olarak da	0.00	0.00	0.00	1	0	0	0.00	0.00
58	yine de de	0.00	0.00	0.00	0	0	0	0.00	0.00
59		0.00	0.00	0.00	0	0	0	0.00	0.00
Average F-measure				0.66					
Weighted Average F-measure				0.81					
Calculated F-measure				0.73					

Table 4.12: Performance of model in row #13 in Table 4.2 for DCs not in the Turkish TCL found in the test set with their frequency and entropy per DC.



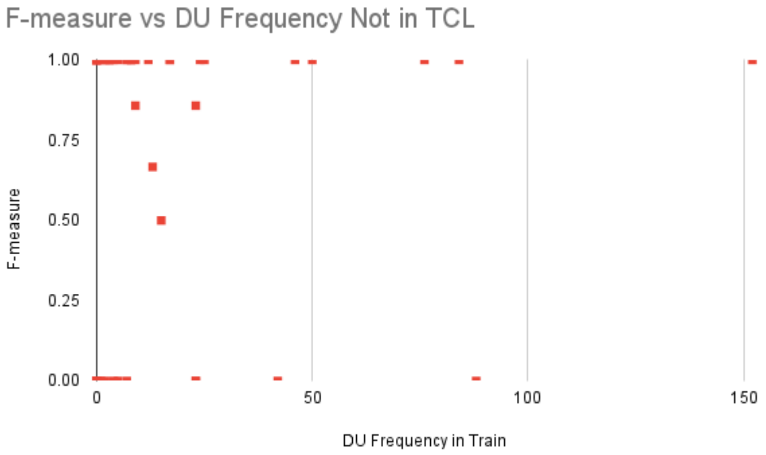
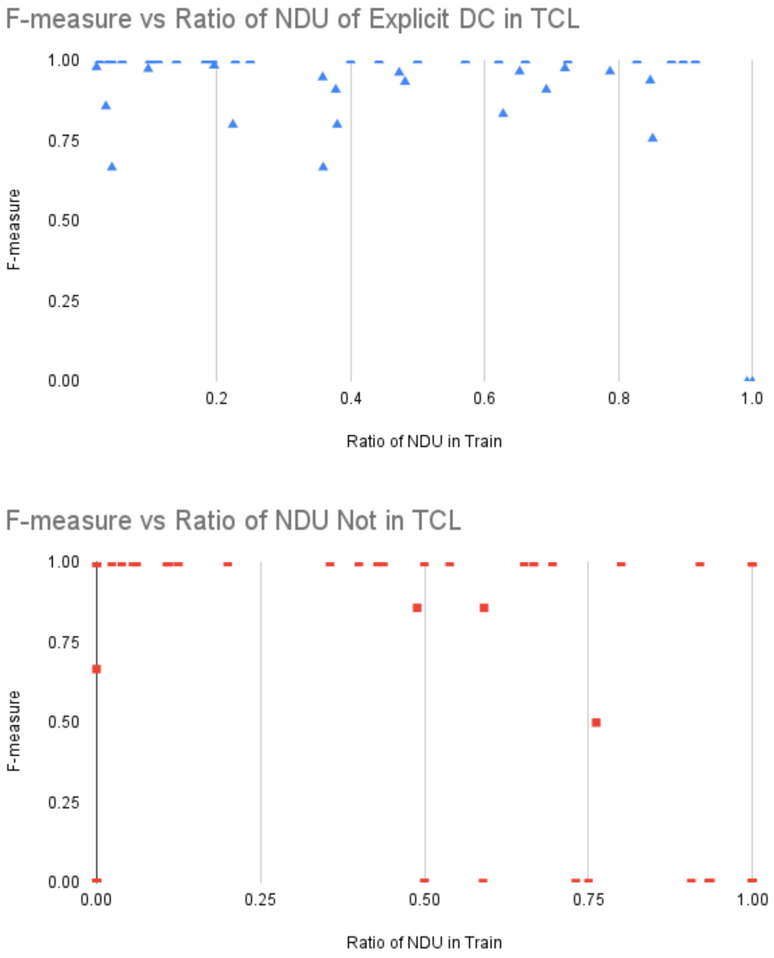


Figure 4.4: Frequency of Turkish DUs in training set of each DC vs F-measure for TDB test set.


F-measure vs Ratio of NDU of Explicit DC in TCL

F-measure vs Ratio of NDU Not in TCL

Figure 4.5: Ratio of Turkish NDUs of each DC vs F-measure for TDB test set.

of correlation.

Figure 4.5 shows the ratio of NDU (frequency of NDU / frequency of NDU + frequency of DU) for the DC in the training set versus the F-measure of that DC in the test set. Again, a higher ratio means that a DC is likely to be NDU, but the model seems to be able to correctly annotate DCs as DUs even if they have a high NDU ratio.

Figure 4.6 shows the entropy of the DC in the training set versus the F-measure of that DC in the test set. Interestingly, it seems that most DCs in Turkish are somewhat ambiguous (many data points are located towards the right of the graph). Yet the model seems to be able to disambiguate a large portion of them.

Figures 4.4, 4.5, and 4.6 seem to show that the DCs found in the TCL are likely to be well

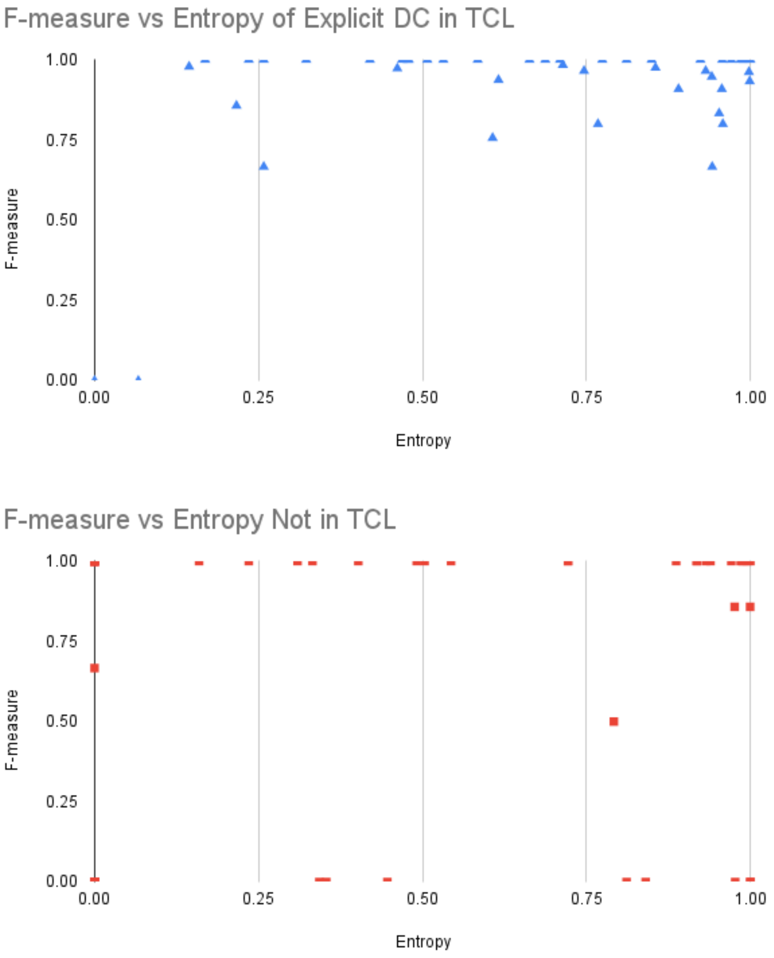


Figure 4.6: Entropy of each Turkish DC vs F-measure for TDB test set.

annotated, while those not found in the TCL are mostly found with an F-measure of 1 or 0. This is likely caused by those particular DCs appearing only once in the test set.

4.2.1.3 Chinese

For Chinese we used two lexicons of DCs: Han yu guan lian ci ci dian (王起澜et al., 1989) henceforth CDCL1 and Guan lian ci yu ci dian (戴木金et al., 1988) henceforth CDCL2. CDCL1 contains 400 phrases that can be part of a Chinese DC; whereas CDCL2 contains 321 phrases. The intersection of the two contains 224 phrases and, therefore, the union of the two lexicon contains 497 phrases. Similarly to English and Turkish, with these lexicons, we can analyse the performance of each phrase in the test set. Of the 497 phrases in the union, only 63 are found in the test set of the DISRPT 2021, and only 50 of them are common in the two lexicons. The model used for the analysis is model 4 in row #1 in Table 4.3.

Inventory 1: CDCL1

Table 4.13 shows the performance of the DCs in the CDCL1. The average F-measure is 88.90, the weighted F-measure is 94.09, and the calculated F-measure is 93.79. The correlation between the F-measure and the frequency of DU in the training set is 0.1618, between the F-measure and the ratio of NDU in the training set is 0.0292 and between the F-measure and the entropy of the training set is 0.2039. These indicate that these features are not correlated to the F-measure for this lexicon. Figures 4.7, 4.8, and 4.9 show this visually.

Inventory 2: CDCL2

Table 4.14 shows the performance of the DCs in the CDCL2. The average F-measure is 86.11, the weighted F-measure is 93.54, and the calculated F-measure is 93.37. These performances are slightly lower then with the CDCL1, although the test set contains 58 phrases in the CDCL2 compared to 55 in the CDCL1. For the frequency of DU in the training set the correlation is 0.1457; for the ratio of NDU it is -0.0576, and for entropy it is 0.0744. Again this shows that these features do

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
1	因	1.00	1.00	1.00	2	1	1	0.50	1.00
2	之所以	1.00	1.00	1.00	1	1	1	0.50	1.00
3	才	1.00	1.00	1.00	1	3	3	0.50	1.00
4	特别是	1.00	1.00	1.00	2	7	8	0.53	1.00
5	如	1.00	1.00	1.00	3	9	11	0.55	0.99
6	时	1.00	1.00	1.00	5	18	30	0.63	0.95
7	却	1.00	1.00	1.00	1	6	3	0.33	0.92
8	为了	1.00	1.00	1.00	5	8	4	0.33	0.92
9	以	1.00	1.00	1.00	2	16	94	0.85	0.60
10	为	1.00	1.00	1.00	2	28	235	0.89	0.49
11	如果	1.00	1.00	1.00	7	9	1	0.10	0.47
12	一	1.00	1.00	1.00	1	9	271	0.97	0.21
13	同时	1.00	1.00	1.00	3	39	0	0.00	0.00
14	至于	1.00	1.00	1.00	1	0	0	0.00	0.00
15	其实	1.00	1.00	1.00	1	0	1	1.00	0.00
16	虽然	1.00	1.00	1.00	2	3	0	0.00	0.00
17	或	1.00	1.00	1.00	1	0	41	1.00	0.00
18	不仅	1.00	1.00	1.00	7	4	0	0.00	0.00
19	只是	1.00	1.00	1.00	1	2	0	0.00	0.00
20	不过	1.00	1.00	1.00	3	2	0	0.00	0.00
21	但是	1.00	1.00	1.00	2	8	0	0.00	0.00
22	假如	1.00	1.00	1.00	1	2	0	0.00	0.00
23	而且	1.00	1.00	1.00	4	12	0	0.00	0.00
24	那	1.00	1.00	1.00	1	0	1	1.00	0.00
25	首先	1.00	1.00	1.00	2	2	0	0.00	0.00
26	其次	1.00	1.00	1.00	1	1	0	0.00	0.00
27	此外	1.00	1.00	1.00	3	25	0	0.00	0.00
28	所以	1.00	1.00	1.00	4	1	0	0.00	0.00
29	既然	1.00	1.00	1.00	1	0	0	0.00	0.00
30	假使	1.00	1.00	1.00	1	0	0	0.00	0.00
31	即便	1.00	1.00	1.00	1	0	0	0.00	0.00
32	的话	1.00	1.00	1.00	1	0	0	0.00	0.00
33	那么	1.00	1.00	1.00	2	1	0	0.00	0.00
34	尽管	1.00	1.00	1.00	8	8	0	0.00	0.00
35	随着	1.00	1.00	1.00	5	22	0	0.00	0.00
36	不管	1.00	1.00	1.00	1	1	0	0.00	0.00
37	只要	1.00	1.00	1.00	1	3	0	0.00	0.00
38	即使	1.00	1.00	1.00	1	1	0	0.00	0.00
39	若	1.00	1.00	1.00	1	7	0	0.00	0.00
40	因此	1.00	1.00	1.00	1	7	0	0.00	0.00
41	但	0.95	1.00	0.98	20	34	0	0.00	0.00
42	而	0.92	1.00	0.96	11	24	26	0.52	1.00
43	后	0.91	1.00	0.95	10	30	23	0.43	0.99
44	则	0.91	1.00	0.95	10	17	3	0.15	0.61
45	并	1.00	0.85	0.92	13	76	22	0.22	0.77
46	由于	0.83	1.00	0.91	5	13	0	0.00	0.00
47	还	0.89	0.89	0.89	9	59	20	0.25	0.82
48	也	0.84	0.91	0.88	35	68	21	0.24	0.79
49	又	1.00	0.67	0.80	3	12	10	0.45	0.99
50	就	1.00	0.50	0.67	4	3	51	0.94	0.31
51	即	0.00	0.00	0.00	0	5	5	0.50	1.00
52	更	0.00	0.00	0.00	0	2	21	0.91	0.43
53	是	0.00	0.00	0.00	0	25	276	0.92	0.41
54	另	0.00	0.00	0.00	1	0	2	1.00	0.00
55	可	0.00	0.00	0.00	1	0	34	1.00	0.00
Average F-measure				0.89					
Weighted Average F-measure				0.94					
Calculated F-measure				0.94					

Table 4.13: Performance of model for Chinese in row #1 in Table 4.3 for CDCL1 DCs found in the test set with their frequency and entropy per DC.

not correlate well with the performances of that DC, Figures 4.7, 4.8, and 4.9 show this visually.

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
1	因	1.00	1.00	1.00	2	1	1	0.50	1.00
2	之所以	1.00	1.00	1.00	1	1	1	0.50	1.00
3	才	1.00	1.00	1.00	1	3	3	0.50	1.00
4	特别是	1.00	1.00	1.00	2	7	8	0.53	1.00
5	如	1.00	1.00	1.00	3	9	11	0.55	0.99
6	却	1.00	1.00	1.00	1	6	3	0.33	0.92
7	为了	1.00	1.00	1.00	5	8	4	0.33	0.92
8	以	1.00	1.00	1.00	2	16	94	0.85	0.60
9	以及	1.00	1.00	1.00	1	4	31	0.89	0.51
10	如果	1.00	1.00	1.00	7	9	1	0.10	0.47
11	其中	1.00	1.00	1.00	3	75	4	0.05	0.29
12	一	1.00	1.00	1.00	1	9	271	0.97	0.21
13	同时	1.00	1.00	1.00	3	39	0	0.00	0.00
14	至于	1.00	1.00	1.00	1	0	0	0.00	0.00
15	其实	1.00	1.00	1.00	1	0	1	1.00	0.00
16	虽然	1.00	1.00	1.00	2	3	0	0.00	0.00
17	或	1.00	1.00	1.00	1	0	41	1.00	0.00
18	不仅	1.00	1.00	1.00	7	4	0	0.00	0.00
19	结果	1.00	1.00	1.00	2	0	3	1.00	0.00
20	只是	1.00	1.00	1.00	1	2	0	0.00	0.00
21	不过	1.00	1.00	1.00	3	2	0	0.00	0.00
22	但是	1.00	1.00	1.00	2	8	0	0.00	0.00
23	假如	1.00	1.00	1.00	1	2	0	0.00	0.00
24	而且	1.00	1.00	1.00	4	12	0	0.00	0.00
25	那	1.00	1.00	1.00	1	0	1	1.00	0.00
26	故	1.00	1.00	1.00	1	2	0	0.00	0.00
27	首先	1.00	1.00	1.00	2	2	0	0.00	0.00
28	其次	1.00	1.00	1.00	1	1	0	0.00	0.00
29	此外	1.00	1.00	1.00	3	25	0	0.00	0.00
30	所以	1.00	1.00	1.00	4	1	0	0.00	0.00
31	既然	1.00	1.00	1.00	1	0	0	0.00	0.00
32	假使	1.00	1.00	1.00	1	0	0	0.00	0.00
33	即便	1.00	1.00	1.00	1	0	0	0.00	0.00
34	的话	1.00	1.00	1.00	1	0	0	0.00	0.00
35	那么	1.00	1.00	1.00	2	1	0	0.00	0.00
36	尽管	1.00	1.00	1.00	8	8	0	0.00	0.00
37	不管	1.00	1.00	1.00	1	1	0	0.00	0.00
38	只要	1.00	1.00	1.00	1	3	0	0.00	0.00
39	即使	1.00	1.00	1.00	1	1	0	0.00	0.00
40	相反	1.00	1.00	1.00	1	0	0	0.00	0.00
41	若	1.00	1.00	1.00	1	7	0	0.00	0.00
42	因此	1.00	1.00	1.00	1	7	0	0.00	0.00
43	但	0.95	1.00	0.98	20	34	0	0.00	0.00
44	而	0.92	1.00	0.96	11	24	26	0.52	1.00
45	则	0.91	1.00	0.95	10	17	3	0.15	0.61
46	并	1.00	0.85	0.92	13	76	22	0.22	0.77
47	由于	0.83	1.00	0.91	5	13	0	0.00	0.00
48	还	0.89	0.89	0.89	9	59	20	0.25	0.82
49	也	0.84	0.91	0.88	35	68	21	0.24	0.79
50	又	1.00	0.67	0.80	3	12	10	0.45	0.99
51	就	1.00	0.50	0.67	4	3	51	0.94	0.31
52	即	0.00	0.00	0.00	0	5	5	0.50	1.00
53	更	0.00	0.00	0.00	0	2	21	0.91	0.43
54	是	0.00	0.00	0.00	0	25	276	0.92	0.41
55	加上	0.00	0.00	0.00	1	0	1	1.00	0.00
56	可	0.00	0.00	0.00	1	0	34	1.00	0.00
57	和	0.00	0.00	0.00	1	0	545	1.00	0.00
58	每	0.00	0.00	0.00	0	0	34	1.00	0.00
Average F-measure				0.86					
Weighted Average F-measure				0.94					
Calculated F-measure				0.93					

Table 4.14: Performance of model for Chinese in row #1 in Table 4.3 for CDCL2 DCs found in the test set with their frequency and entropy per DC.

Inventory 3: Union of CDCL1 and CDCL2

The union of the the two lexicons results in an average F-measure of 85.55, weighted F-measure of 93.46, and calculated F-measure of 93.39. The only correlation worth noting is that of the ratio of NDU in the training set, which is -0.5590, indicating a slight correlation; whereas the other features are not correlated similarly to the CDCL1 and the CDCL2.

Inventory 4: Intersection of CDCL1 and CDCL2

For the intersection of CDCL1 and CDCL2, we have an average F-measure of 89.89, a weighted F-measure of 94.15, and a calculated F-measure of 93.81. This is to be expected, as the intersection likely contains DCs that are less ambiguous. None of the correlations are noteworthy, as they are similar to those observed for the CDCL1 and the CDCL2.

Inventory 5: Not in CDCL1 or CDCL2

Finally, we analysed the DCs that are in the DISRPT 2021 test set and are not found in either the CDCL1 or the CDCL2, of which there are 65. The average F-measure is 49.48 (due to many DCs with a zero F-measure), a weighted F-measure of 71.63, and a calculated F-measure of 74.85. It is clear that these DCs are more difficult for the model to identify. Table 4.15 shows the performance of individual DCs, most of them appearing only once or twice in the test set. Figures 4.7, 4.8, and 4.9 show the correlation between their F-measure and the same 3 features (frequency DU, ratio NDU, and entropy); whose values are 0.3163, -0.0724, 0.5029, respectively. Only the entropy shows a slight correlation.

Figure 4.7 shows the frequency of DU for each DC in the training set versus its F-measure in the test set. Similarly to Turkish and English, we expected DCs with more DU annotations in the training set to have a better F-measure on the test set. However, this is not really the case.

Figure 4.8 shows the ratio of NDU for the DCs in the training set versus the F-measure of that DC in the test set. The ratio of NDU should have a negative correlation with the F-measure, because as stated before the ratio of NDU is a feature that signals that a DC is more likely to be used in NDU, but again even those with a NDU ratio of close to 1 have an F-measure of 1. This seems to indicate

Nb	DC	Precision	Recall	F-measure	DU Freq Test	DU Freq Train	NDU Freq Train	NDU ratio Train	Entropy Train
1	不久	1.00	1.00	1.00	1	1	1	0.50	1.00
2	继	1.00	1.00	1.00	1	3	3	0.50	1.00
3	为此	1.00	1.00	1.00	2	2	2	0.50	1.00
4	自	1.00	1.00	1.00	1	12	15	0.56	0.99
5	下	1.00	1.00	1.00	5	9	14	0.61	0.97
6	之后	1.00	1.00	1.00	2	3	5	0.63	0.95
7	以来	1.00	1.00	1.00	2	19	34	0.64	0.94
8	使	1.00	1.00	1.00	3	33	18	0.35	0.94
9	通过	1.00	1.00	1.00	1	14	27	0.66	0.93
10	这样	1.00	1.00	1.00	2	2	4	0.67	0.92
11	经过	1.00	1.00	1.00	2	8	3	0.27	0.85
12	不再	1.00	1.00	1.00	1	1	3	0.75	0.81
13	二	1.00	1.00	1.00	1	8	28	0.78	0.76
14	令	1.00	1.00	1.00	1	1	6	0.86	0.59
15	三	1.00	1.00	1.00	1	5	82	0.94	0.32
16	从	1.00	1.00	1.00	1	1	70	0.99	0.11
17	当	1.00	1.00	1.00	3	6	0	0.00	0.00
18	本来	1.00	1.00	1.00	1	0	0	0.00	0.00
19	前	1.00	1.00	1.00	1	0	39	1.00	0.00
20	从而使	1.00	1.00	1.00	1	0	0	0.00	0.00
21	的同时	1.00	1.00	1.00	2	7	0	0.00	0.00
22	就证明	1.00	1.00	1.00	1	0	0	0.00	0.00
23	与此同时	1.00	1.00	1.00	2	6	0	0.00	0.00
24	如果说	1.00	1.00	1.00	1	0	0	0.00	0.00
25	的过程中	1.00	1.00	1.00	1	0	0	0.00	0.00
26	之际	1.00	1.00	1.00	1	1	0	0.00	0.00
27	以使	1.00	1.00	1.00	1	1	0	0.00	0.00
28	这表明	1.00	1.00	1.00	1	2	0	0.00	0.00
29	的时候	1.00	1.00	1.00	2	1	0	0.00	0.00
30	就在	1.00	1.00	1.00	1	0	0	0.00	0.00
31	考虑到	1.00	1.00	1.00	1	0	1	1.00	0.00
32	在	1.00	0.75	0.86	16	44	475	0.92	0.42
33	使得	0.67	1.00	0.80	2	1	0	0.00	0.00
34	造成	0.00	0.00	0.00	0	1	6	0.86	0.59
35	间	0.00	0.00	0.00	1	1	25	0.96	0.24
36	中	0.00	0.00	0.00	1	5	183	0.97	0.18
37	这	0.00	0.00	0.00	1	4	152	0.97	0.17
38	的	0.00	0.00	0.00	0	13	2233	0.99	0.05
39	却也	0.00	0.00	0.00	1	1	0	0.00	0.00
40	进一步	0.00	0.00	0.00	1	0	49	1.00	0.00
41	每逢	0.00	0.00	0.00	1	0	0	0.00	0.00
42	这是导致	0.00	0.00	0.00	1	0	0	0.00	0.00
43	的主要因素	0.00	0.00	0.00	1	0	1	1.00	0.00
44	等因素	0.00	0.00	0.00	1	0	0	0.00	0.00
45	之前	0.00	0.00	0.00	1	0	1	1.00	0.00
46	更引人注目的是	0.00	0.00	0.00	1	0	0	0.00	0.00
47	所有信息显示	0.00	0.00	0.00	1	0	0	0.00	0.00
48	是因为	0.00	0.00	0.00	1	0	2	1.00	0.00
49	的缘故	0.00	0.00	0.00	1	0	0	0.00	0.00
50	大概是由于	0.00	0.00	0.00	1	0	0	0.00	0.00
51	这是	0.00	0.00	0.00	1	0	23	1.00	0.00
52	的原因	0.00	0.00	0.00	1	0	3	1.00	0.00
53	举例	0.00	0.00	0.00	1	0	0	0.00	0.00
54	这实际上是	0.00	0.00	0.00	1	0	0	0.00	0.00
55	初期	0.00	0.00	0.00	1	0	3	1.00	0.00
56	但也	0.00	0.00	0.00	1	0	0	0.00	0.00
57	到	0.00	0.00	0.00	1	0	117	1.00	0.00
58	不仅仅	0.00	0.00	0.00	0	0	0	0.00	0.00
59		0.00	0.00	0.00	0	0	0	0.00	0.00
60	可以说	0.00	0.00	0.00	0	0	2	1.00	0.00
61	应该说	0.00	0.00	0.00	0	0	0	0.00	0.00
62	也是因为	0.00	0.00	0.00	0	0	0	0.00	0.00
63	缘故	0.00	0.00	0.00	0	0	0	0.00	0.00
64	大概	0.00	0.00	0.00	0	0	0	0.00	0.00
65	均	0.00	0.00	0.00	0	0	27	1.00	0.00
66	就要求	0.00	0.00	0.00	0	0	0	0.00	0.00
Average F-measure				0.49					
Weighted Average F-measure				0.72					
Calculated F-measure				0.75					

Table 4.15: Performance of model for Chinese in row #1 in Table 4.3 for DCs not in the CDCL1 or CDCL2 found in the test set with their frequency and entropy per DC.

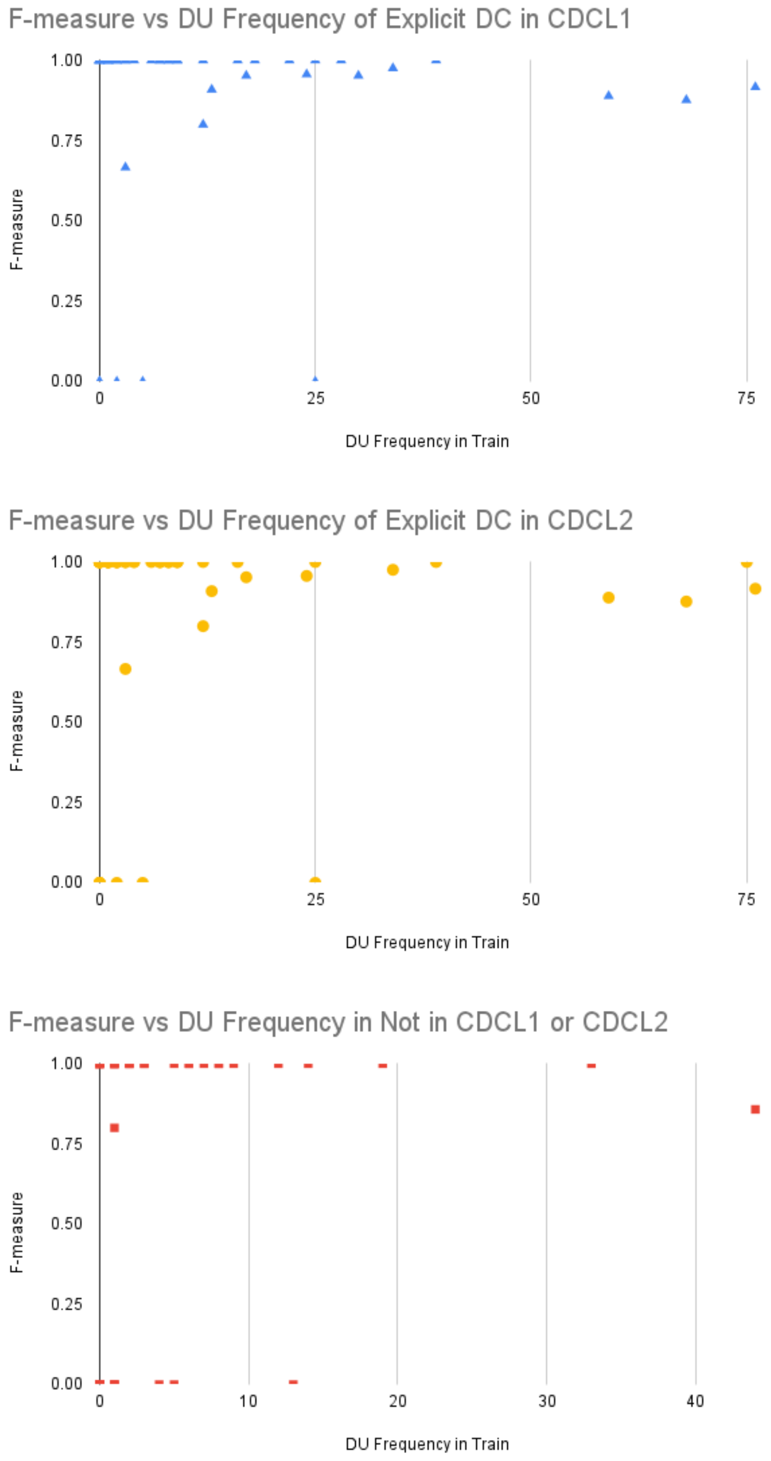


Figure 4.7: Frequency of DUs in training set of each DC vs F-measure for CDTB test set.

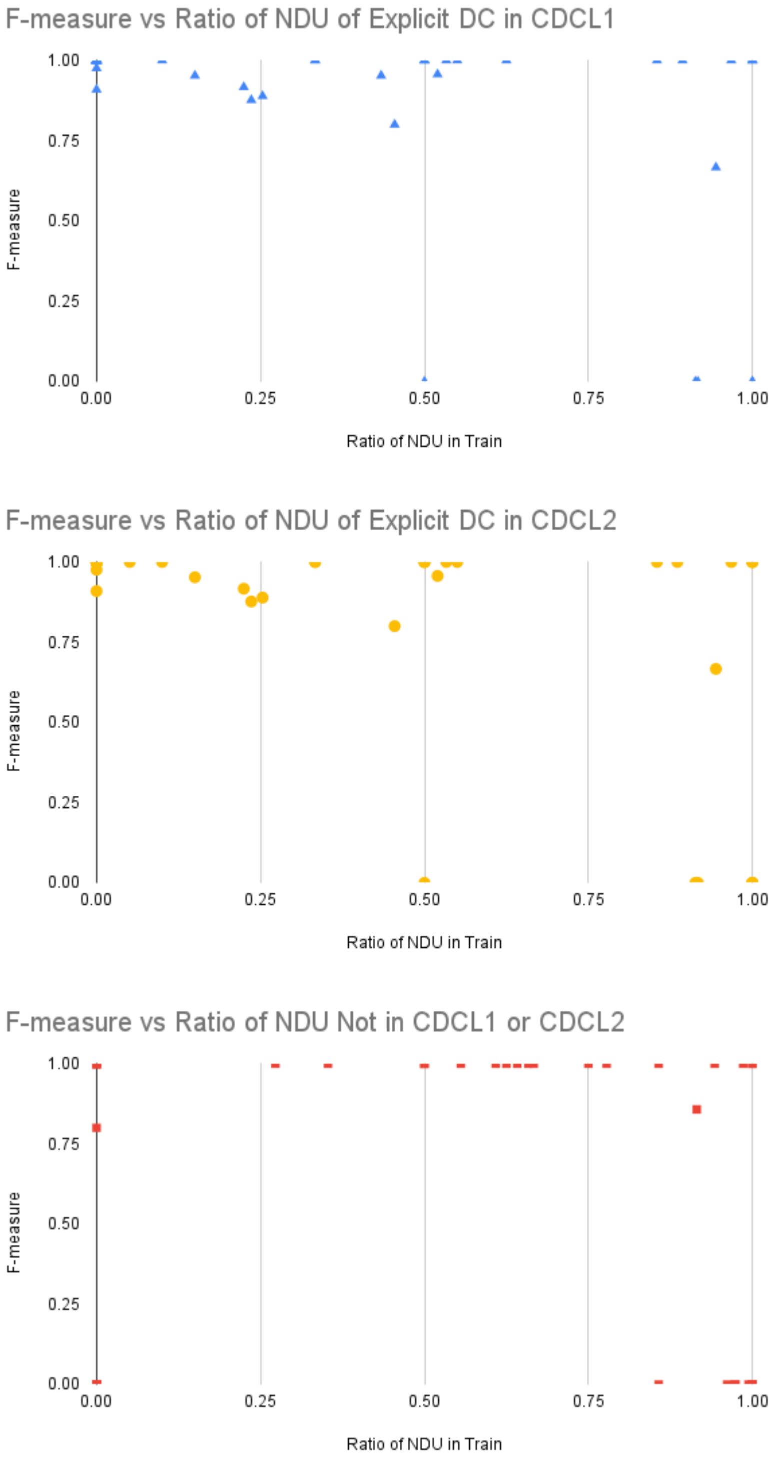


Figure 4.8: Ratio of NDUs of each DC vs F-measure for CDTB test set.

that the model can identify a DC in DU even if that DC appears mostly in NDU form in the training set.

Figure 4.9 shows the entropy of the DCs in the training set versus their F-measure in the test set. Being a measure of ambiguity, most DC in Chinese appear to be ambiguous, yet the model is able to identify them correctly.

Figures 4.7, 4.8, 4.9 seem to indicate a difference in performance based on whether that DC is common enough to appear in a DC lexicon. Most likely because these DCs are `Explicit` while the others are `AltLex`.

4.2.2 Analysis of The Synthetic Data Sets

The second investigation is aimed at identifying errors in our synthetic datasets. We performed this analysis by inspecting the English annotations in the parallel sentences.

4.2.2.1 Turkish

Synthetic Projection Data Set (TUR-PJ)

We analysed the synthetic Turkish projection data set (TUR-PJ) by inspecting the English parallel sentences to see what kind of DCs have been projected. Figures 4.10, 4.11, and 4.12 show various errors that seem to be quite commonly found in the projected data set.

Figure 4.10 shows the English DC *also* annotated with `B-Conn` being projected onto a comma in the Turkish parallel sentence. This is incorrect and may cause the model to learn something that is not valuable. Furthermore, we can also find instances where a comma in the Turkish dataset is annotated with a `I-Conn`. In total, the error can be observed 527 times; 514 times for a comma annotated as `B-Conn`, and 13 times where a comma is annotated as `I-Conn`. For a data set with 4,166 tokens annotated as DCs, this is not insignificant. A simple heuristic that filters this type of projection could be very beneficial here.

Figure 4.11 shows *also* which in this sentence is not a DC, although the English model annotated it as such. Therefore, the projection of the annotated NDU is also incorrect. This error most likely

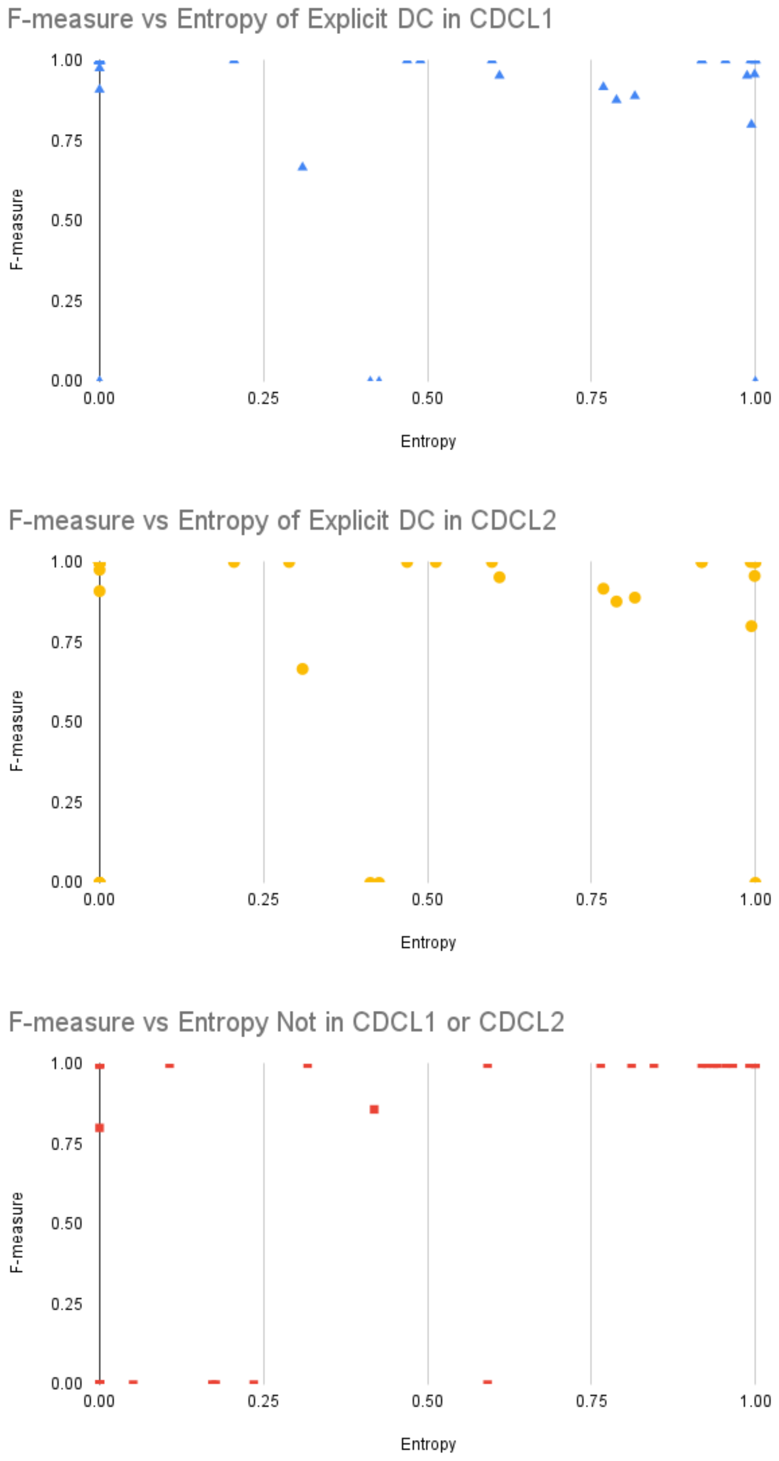


Figure 4.9: Entropy of each DC vs F-measure for CDTB test set.

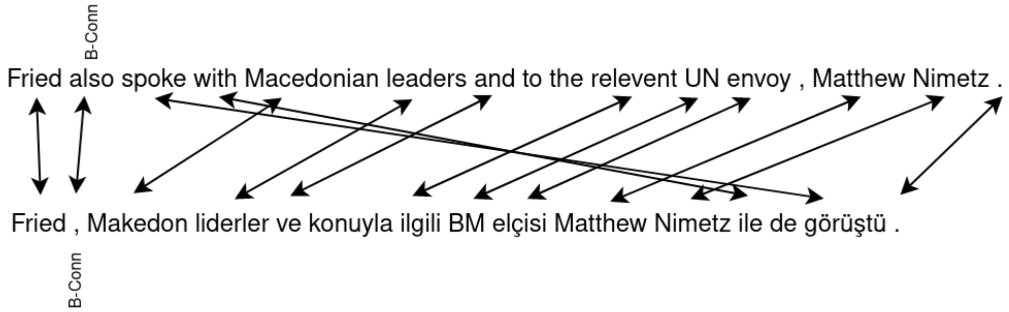

B-Conn
Fried also spoke with Macedonian leaders and to the relevent UN envoy , Matthew Nimetz .
Fried , Makedon liderler ve konuyla ilgili BM elçisi Matthew Nimetz ile de görüştü .
B-Conn

Figure 4.10: Example of error for Turkish projection 1: DC *also* annotation is projected onto a comma which is not a valid DC.

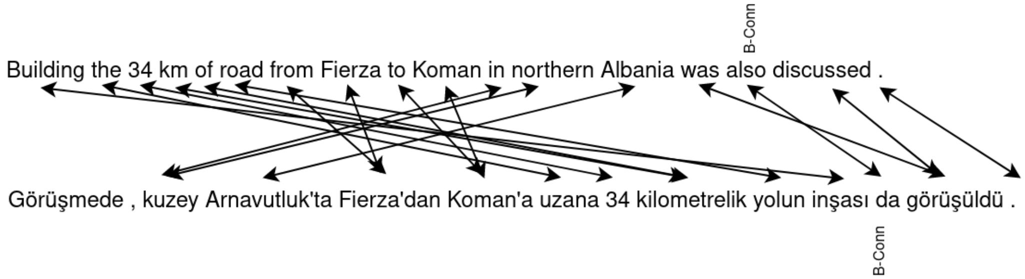

B-Conn
Building the 34 km of road from Fierza to Koman in northern Albania was also discussed .
Görüşmede , kuzey Arnavutluk'ta Fierza'dan Koman'a uzana 34 kilometrelik yolun inşası da görüşüldü .
B-Conn

Figure 4.11: Example of error for Turkish projection 2: DC *also* is in NDU, therefore projecting an incorrectly annotated annotation to Turkish counterpart.

occurs often, but the exact number of occurrences can only be determined by closely verifying each instance in the dataset, which is time consuming and requires linguistic expertise.

Firgure 4.12 shows the DC *however* being dropped from the English to the Turkish parallel data, yet the sentence is kept because the projection algorithm (see Section 3.1) allows for sentences with NDU to be added to the corpus. We counted 823 sentences where a DC in English was dropped in the Turkish parallel sentence; this is quite a significant portion of the dataset. A heuristic that would not allow this kind of projection to occur could be used to avoid these errors.

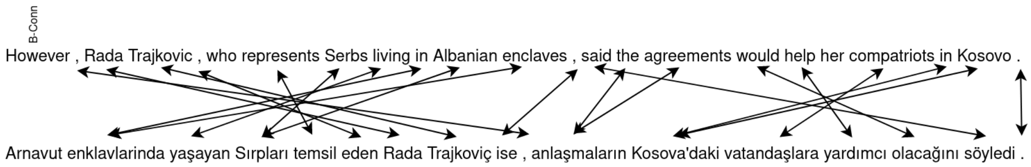

B-Conn
However , Rada Trajkovic , who represents Serbs living in Albanian enclaves , said the agreements would help her compatriots in Kosovo .
Arnavut enklavlarinda yaşayan Sırpları temsil eden Rada Trajkoviç ise , anlaşmaların Kosova'daki vatandaşlara yardımcı olacağını söyledi .

Figure 4.12: Example of error for Turkish projection 3: DC lost in translation.

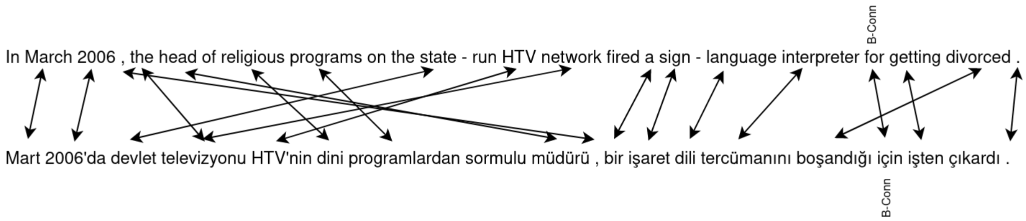

B-Conn
In March 2006 , the head of religious programs on the state - run HTV network fired a sign - language interpreter for getting divorced .
Mart 2006'da devlet televizyonu HTV'nin dini programlardan sormulu müdürü , bir işaret dili tercümanını boşandığı için işten çıkardı .
B-Conn

Figure 4.13: Example of error for Turkish agreement 1: DC *for* annotated but should not be, in NDU.

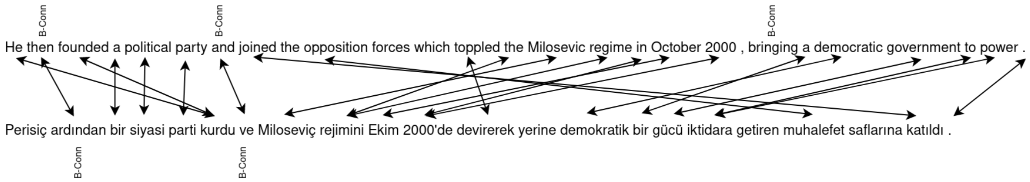

B-Conn
B-Conn
B-Conn
He then founded a political party and joined the opposition forces which toppled the Milosevic regime in October 2000 , bringing a democratic government to power .
Perisiç ardından bir siyasi parti kurdu ve Miloseviç rejimini Ekim 2000'de devirerek yerine demokratik bir gücü iktidara getiren muhalefet saflarına katıldı .
B-Conn
B-Conn

Figure 4.14: Example of error for Turkish agreement 2: DC *bringing* is annotated but should not be, and is lot in translation.

Synthetic Agreement Data Set (TUR-AG)

We tried to visually analyse the synthetic Turkish agreement dataset (TUR-AG), similarly to what we did for the projection dataset (TUR-PJ). We were able to find errors where DC terms are annotated but they are in fact in NDU (see Figure 4.13). It is likely that this type of error is the most frequent, as DCs that are ambiguous in English are likely to also be ambiguous in the target language. Figure 4.14 shows a different type of error where the term annotated is not a DC and was lost in translation. It stands to reason that this type of error could happen in the reverse direction, where the Turkish model annotated a term that is not a DC in the Turkish corpus, and in the English corpus that term was dropped in translation. In other words, the Turkish corpus would contain a DC that only the Turkish model identified, because the English model did not know that it existed. This type of error would result in a corpus that reinforces the errors of the model. A possible solution would be to employ an ensemble of models to do the annotation in both the source and target language.

We initially believed that the main source of error from in TUR-AG would come from the use of SimAlign (see Section 3.3.2.2) to create the word alignments, causing error accumulation downstream. To test this hypothesis, we created two new agreement data sets: using a more lenient word alignment with a threshold probability of 75% and using a more strict alignment set at 90%, recall from Section 3.3.2.2 that the datasets used in the previous experiments uses 85% as a threshold. We trained model 4 (BERT + CRF) described in Section 4.1.2. The dataset with 90% alignment probability has 2589 instances to train on, which resulted in an F-measure of 70.95 (±0.92) on the TDB test set. This is much weaker than the original agreement only model, which had an F-measure of 87.99 (±0.54) (see Table 4.2). However, the model that was trained on the data set with alignment probability of 75%, had 93,572 instances, and resulted in an F-measure of 90.24 (±0.15). This seems to go against our hypothesis that additional data does not benefit the task of DC identification and that the error accumulation from using a word alignment tool is significant.

4.2.2.2 Chinese

Synthetic Projection Data Set (ZHO-PJ)

Similarly to the Turkish projection dataset, the Chinese projection dataset (ZHO-PJ) suffers from a similar problem. Terms are marked as DC in DU when they do not signal any discourse relation. Figure 4.15 shows the word *shaking*, which was annotated by the English model, then projected onto its Chinese counterpart 震荡. This error occurs because of the restrictions of the projection algorithm on which sentences are being projected. Only sentences where no annotation was found in the target sentence can receive an annotation (see Section 2.1.1), which ends up propagating errors that the source model makes on the target language dataset. Another error that this dataset shares with the Turkish projection is the annotated NDUs. Figure 4.16 shows *specifically* which was incorrectly annotated by the English model; again, it is projected onto its Chinese counterpart 明确. These types of errors are hard to automatically determine and require a closer inspection of the resulting dataset to count how often they occurs.

The error where commas are marked with DC annotation that was found in the Turkish projection dataset does happen in Chinese as well, however, fewer times; occurring only 33 times in the Chinese projection dataset. This may be because the word alignment used for Chinese synthetic datasets are manual gold standard (see Section 3.3.2.1), which means that fewer commas are aligned with words. However, 1678 sentences of the dataset have terms that were annotated by the English

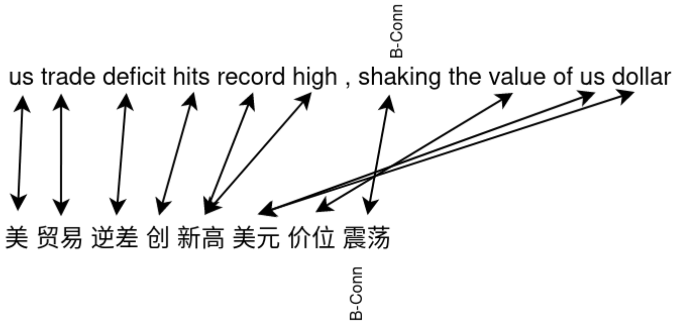

B-Conn
us trade deficit hits record high , shaking the value of us dollar
美 贸易 逆差 创 新高 美元 价位 震荡
B-Conn

Figure 4.15: Example of error For Chinese projection 1: DC *shaking* is annotated but is not a DC, and is projected onto the Chinese parallel phrase.

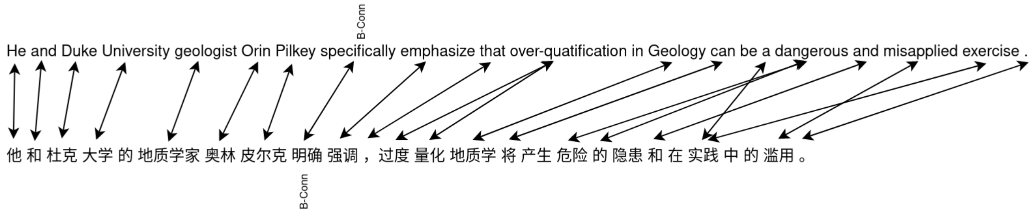

B-Conn
He and Duke University geologist Orin Pilkey specifically emphasize that over-quatification in Geology can be a dangerous and misapplied exercise .
他 和 杜克 大学 的 地质学家 奥林 皮尔克 明确 强调 , 过度 量化 地质学 将 产生 危险 的 隐患 和 在 实践 中 的 滥用 。
B-Conn

Figure 4.16: Example of error for Chinese projection 2: DC *specifically* is annotated but is NDU, and is projected onto the Chinese parallel phrase.

model and that term is dropped in the Chinese translation. resulting in Chinese sentences that have no terms with no DC annotation. Similarly to Turkish, the best solution would be not to keep these sentences.

Synthetic Agreement Data Set (ZHO-AG)

Unlike the synthetic Turkish agreement dataset (TUR-AG), we did not use a word alignment tool (see Section 3.3.2.1) to generate ZHO-AG. This means that in principle there should be no accumulation of errors from the word alignment in the synthetic DC identification. Additionally, the model trained on the Chinese agreement data set performs almost as well as the model trained on the CDTB, making it difficult to identify errors in the synthetic data sets.

In this chapter we have measured the performance of each of our models on the task of DC identification using the DISRPT 2021 evaluation script. We have compared our best models for each language (English, Chinese, Turkish) with the results of the participants at DISRPT 2021. The performance of our models are comparable if not better than the SOTA systems. Finally, an analysis of the best models for each language was performed, seeing how well the models perform on specific DCs. We also analysed the errors that can occur in our synthetic datasets due to the way the projection and agreement algorithms were designed. Chapter 5 concludes this thesis and presents future work.

Chapter 5

Conclusion and Future Work

This thesis explored the automatic identification of discourse connectives in a multilingual setting. Our models used transformers as embeddings without the need for heavy linguistic features. We evaluated our work on English, Turkish, and Chinese as part of the DISRPT shared task. Our simple approaches with language-specific BERT embedding seem to perform well, outperforming or matching the best results seen in DISRPT 2021 for English, Turkish, and Chinese. Our experiment show that adding handcrafted features is not beneficial to BERT embeddings. As part of our exploration we also developped two methods for data augmentation based on annotation projection and agreement. However the performance using the synthetic data sets showed that they do not help the models and actually seem to harm their performance. An analysis of the corpora shows that there are several systematic errors in the synthetic data sets and simple heuristics may be used to filter them out.

Our experiments with corpus reduction showed that our main model trained on only 25% of the PDTB or trained on only 25% of the TDB outperformed most teams at DISRPT 2021. This seems to indicate that low-resourced languages do not need the full size corpus of the PDTB (44k) or even the TDB (25k) to perform well (over 90 F-measure) on the task DC identification. For Chinese, this would mean roughly doubling the size of the CDTB. We believe this is because BERT-like models already learn some of discourse information in the pre-training tasks, and fine-tuning on a DC identification task simply gives more importance to the attention head in the multi-head attention that is responsible for this type of information.

Our experiments with cross-language training using multilingual BERT (see Section 4.1) showed poorer performance compared to training on a specific language. These results are in line with what Virtanen et al. (2019) found. Indeed, the authors noticed that for Finnish, multilingual BERT under-performed compared to a BERT embedding pre-trained solely on Finnish on a variety of NLP tasks (POS tagging, named entity recognition, dependency parsing, and text classification). In our case, we suspect that this is because the attention heads are responsible for both identifying the language in question and tagging the DC. However, balancing the datasets leads to the lowest resourced language gaining a small amount of performance over not doing so.

An analysis of our best performing models for each language (see Section 4.2) shows that the BERT models can easily identify `Explicit` DCs that can be found in reference lexicons of each language. However, the identification of `AltLex` DCs or DCs not found in the lexicons seems to be more difficult in all languages. This result is likely due to the lack of training examples for these types of DCs. Adding instances for each of these DCs in both DU and NDU may alleviate the performance deficit, although because `AltLex` are open class, obtaining training corpora that covers all of them in sufficient number may be difficult itself.

5.1 Contributions

This thesis presented a number of theoretical and practical contributions, including the following:

(1) The implementation of various DC identification models using transformers as a embedding that do not need any handcrafted features (see Chapter 3). These models achieve SOTA performance of 93.12 F-measure for English, 94.42 F-measure for Turkish and 87.47 F-measure for Chinese.

(2) The creation of synthetic Turkish and Chinese corpora with discourse connective annotation (see Chapter 3). These corpora will be publicly available on the CLaC Github.

(3) Experimentation with the synthetic Turkish and Chinese corpus, in order to augment the data available for those languages and to see if performance is impacted (see Chapter 4). From our

experiments it seems that our synthetic corpora negatively impact model performance.

(4) Experimentation with corpus reduction to determine how much training data is needed to create models that still perform well (see Chapter 4). From our experiments it seems that 5k to 10k training examples can achieve F-measures of 90% in English and Turkish.

(5) An analysis of the models to understand what DCs it identifies well and what DCs it struggles with (see Chapter 4). Our analysis showed that our models struggle with `AltLex` connectives, and that the entropy of a DC is not a good indication of how well that DC will perform.

(6) An analysis of the synthetic Turkish and Chinese corpora and the proposal of simple heuristics to improve them (see Chapter 4). Our analysis revealed several systemic errors that most likely lead to poorer performances.

Contributions 1 to 3 above are the focus of our recently published paper at the 27^{th} International Conference on Natural Language & Information Systems (NLDB-2022) (Muermans and Kosseim, 2022).

5.2 Future Work

In Chapter 3, we presented two methods for creating synthetic data (annotation projection and annotation agreement). Although the methodology is intuitively sound, these corpora did not lead to expected improvements in performance. We speculate that this maybe due to two reasons: (1) BERT is enough (2) the corpora contain many errors. To verify the second hypothesis, we performed a manual inspection of the synthetic datasets. In Chapter 4, we discovered several errors, and proposed simple heuristics to filter them. It would be interesting to apply these heuristics and validate if the new synthetic corpora offer a benefit for low-resource languages (i.e. Chinese). The performance of the Chinese agreement data set is only a few points away (1.16%) from the performance after being trained on the CDTB; indicating that after refinements it would likely improve performance.

Our result with data reduction (see Section 4.1) seems to indicate that the BERT embedding already possesses some level of knowledge about `Explicit` DCs. It would be interesting to analyze the attention heads before fine-tuning and after fine-tuning to see which heads contain this information and how fine-tuning affects BERT. This could also help identify why models struggle with AltLex DCs identification.

Our results with the GPT2 model are very weak (see Section 4.1.1), and more research is needed to determine why this model is underperforming. We suspect that this is because the CRF output layer does not send error signals that are useful for the GPT2 embedding, or DC detection is not something it performs well in.

Our results with multilingual cross training (see Section 4.1) are weaker than expected. It may be worth investigating cross-training using languages that are part of the same family (for English a West Germanic language, for Turkish languages in the Ural-Altaic linguistic family, and for Mandarin Chinese a language from Sino-Tibetan family). Additionally, performing the same experiments with a more powerful multilingual model that has more parameters and attention heads may close the gap between the performance differences.

It is clear from our analysis that more work needs to be done to address the performance of less frequent DCs, such as AltLex (see Section 4.2.1). Whether it be by adding more examples in the training set or identifying why the models fail at this. We believe that solving this issue would greatly improve the models performance overall.

Finally, in Chapter 2 we discussed the CDTB and how DCs in the Chinese language are often part of a pair, also known as discontinuous DCs (DDCs) or paired DCs (Huang et al., 2014). Investigating this further and creating models to specifically address these DCs may be useful in improving the overall performance of the Chinese models. This could be done in a 2-step process: initially, a model can be created to identify if a pair of Chinese phrases form a DDC or not. If so, then a second model could be used to identify DDCs in a sentence, which could be used as part of a pipeline to identify DCs in Chinese.

Finally, in this thesis, we explored the identification for all Explicit and AltLex DCs for English, Turkish and Chinese and showed that our simple BERT based approach does well in each. However, it would be interesting to see how our approach fares with other languages that do not enjoy large embeddings such as BERT.

Bibliography

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, USA, May 2015.

Yoshua Bengio, Patrice Simard, Paolo Frasconi, et al. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, March 1994.

Luisa Bentivogli and Emanuele Pianta. Exploiting parallel texts in the creation of multilingual semantically annotated resources: The MultiSemCor Corpus. *Natural Language Engineering*, 11 (3):247–261, 2005.

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, 19 (2):263–311, 1993. URL https://aclanthology.org/J93-2003.

Debopam Das and Maite Taboada. RST Signalling Corpus: A Corpus of Signals of Coherence Relations. *Lang. Resour. Eval.*, 52(1):149–184, mar 2018. ISSN 1574-020X. doi: 10.1007/s10579-017-9383-x. URL https://doi.org/10.1007/s10579-017-9383-x.

Debopam Das, Tatjana Scheffler, Peter Bourgonje, and Manfred Stede. Constructing a Lexicon of English Discourse Connectives. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 360–365, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5042. URL https://aclanthology.org/W18-5042.

Mohammad Reza Davari. Neural Network Approaches to Medical Toponym Recognition. Unpublished, April 2020. URL https://spectrum.library.concordia.ca/id/eprint/986657/.

Işın Demirşahin and Deniz Zeyrek Bozşahin. *Pair Annotation as a Novel Annotation Procedure: The Case of Turkish Discourse Bank*. 2017. URL https://hdl.handle.net/11511/68723.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. Jointly Learning to Align and Translate with Transformer Models. 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 4453–4462, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1453. URL https://aclanthology.org/D19-1453.

Luke Gessler, Shabnam Behzad, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. DisCoDisCo at the DISRPT2021 Shared Task: A System for Discourse Segmentation, Classification, and Connective Detection. *CoRR*, abs/2109.09777, 2021. URL https://arxiv.org/abs/2109.09777.

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

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL https://spacy.io/. Available at https://spacy.io/.

Hen-Hsen Huang, Tai-Wei Chang, Huanpei Chen, and Hsin-Hsi Chen. Interpretation of Chinese Discourse Connectives for Explicit Discourse Relation Recognition. In *25th International Conference on Computational Linguistics (COLING)*, Dublin, Ireland, 2014.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. SimAlign: High Quality Word Alignments without Parallel Training Data using Static and Contextualized Embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, (EMNLP-2020)*, pages 1627–1643, Punta Cana, Dominican Republic, November 2020. URL https://www.aclweb.org/anthology/2020.findings-emnlp.147.

Peter Jansen, Mihai Surdeanu, and Peter Clark. Discourse Complements Lexical Semantics for Non-factoid Answer Reranking. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL-2014)*, pages 977–986, Baltimore, Maryland, 2014.

Anders Johannsen and Anders Søgaard. Disambiguating Explicit Discourse Connectives without Oracles. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing, (IJCNLP 2013)*, pages 997–1001, Nagoya, Japan, October 2013. URL https://aclanthology.org/I13-1134.

Dan Jurafsky and James H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Pearson Prentice Hall, Upper Saddle River, N.J., 2009. ISBN 9780131873216 0131873210. URL http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd_bxgy_b_img_y.

Majid Laali. Inducing Discourse Resources Using Annotation Projection. PhD Thesis, Concordia University, https://spectrum.library.concordia.ca/983791/, November 2017. URL https://spectrum.library.concordia.ca/983791/.

Majid Laali and Leila Kosseim. Improving Discourse Relation Projection to Build Discourse Annotated Corpora. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, (RANLP 2017)*, pages 407–416, Varna, Bulgaria, September 2017. doi: 10.26615/978-954-452-049-6_054. URL https://doi.org/10.26615/978-954-452-049-6_054.

Majid Laali, Andre Cianflone, and Leila Kosseim. The CLaC Discourse Parser at CoNLL-2016. In *Proceedings of the CoNLL-16 shared task*, pages 92–99, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-2013. URL https://aclanthology.org/K16-2013.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

William D. Lewis and Fei Xia. Automatically Identifying Computationally Relevant Typological Features. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, 2008. URL https://aclanthology.org/I08-2093.

Yang Liu and Maosong Sun. Contrastive Unsupervised Word Alignment with Non-Local Features. In *Proceedings of the Twenty-Ninth Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, (AAAI-2015)*, page 2295–2301, 2015. ISBN 0262511290. URL http://arxiv.org/abs/1410.2082.

Yang Liu, Qun Liu, and Shouxun Lin. Log-Linear Models for Word Alignment. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL-2005)*, pages 459–466, Ann Arbor, Michigan, June 2005. doi: 10.3115/1219840.1219897. URL https://aclanthology.org/P05-1057.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Annie Louis, Aravind Joshi, and Ani Nenkova. Discourse indicators for content selection in summarization. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 147–156, 2010.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, June 1993. ISSN 0891-2017.

Eleni Miltsakaki, Aravind Joshi, Rashmi Prasad, and Bonnie Webber. Annotating Discourse Connectives and Their Arguments. In *Proceedings of the Workshop Frontiers in Corpus Annotation at HLT-NAACL 2004*, pages 9–16, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-2703.

Thomas Chapados Muermans and Leila Kosseim. A BERT-Based Approach for Multilingual Discourse Connective Detection. In *27th International Conference on Applications of Natural Language to Information Systems (NLDB-2022)*, València, Spain, 2022.

Philippe Muller, Chloé Braud, and Mathieu Morey. ToNy: Contextual embeddings for accurate multilingual discourse segmentation of full documents. In *Proceedings of the Workshop on Discourse Relation Parsing and Treebanking 2019*, pages 115–124, Minneapolis, MN, June 2019. doi: 10.18653/v1/W19-2715. URL https://aclanthology.org/W19-2715.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 03 2003. ISSN 0891-2017. doi: 10.1162/089120103321337421. URL https://doi.org/10.1162/089120103321337421.

Robert Östling. Word Order Typology through Multilingual Word Alignment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 205–211, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2034. URL https://aclanthology.org/P15-2034.

Jan-Thorsten Peter, Arne Nix, and Hermann Ney. Generating Alignments Using Target Foresight in Attention-Based Neural Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, 108, 06 2017. doi: 10.1515/pralin-2017-0006.

Emily Pitler and Ani Nenkova. Using Syntax to Disambiguate Explicit Discourse Connectives in Text. In *Proceedings of the Association for Computational Linguistics and International Joint*

Conference on Natural Language Processing (ACL-IJCNLP 2009), pages 13–16, Suntec, Singapore, August 2009. URL https://aclanthology.org/P09-2004.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, pages 2961–2968, Marrakech, Morocco, May 2008. URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/754_paper.pdf.

Alec Radford and Karthik Narasimhan. Improving Language Understanding by Generative Pre-Training. 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

Ines Rehbein, Merel Scholman, and Vera Demberg. Annotating discourse relations in spoken language: A comparison of the PDTB and CCR frameworks. 01 2016.

Bilge Say, Deniz Zeyrek, Kemal Oflazer, and Umut Özge. Development of a corpus and a treebank for present-day written Turkish. *Proceedings of the 11th International Conference of Turkish Linguistics (ICTL)*, pages 183–192, 01 2002.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.

Fei Sha and Fernando Pereira. Shallow Parsing with Conditional Random Fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 134–141, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1073445.1073473. URL https://doi.org/10.3115/1073445.1073473.

Jörg Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012)*, page 2214–2218, Istanbul, Turkey, may 2012. ISBN 978-2-9517408-7-7. URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

Jörg Tiedemann. Improving the Cross-Lingual Projection of Syntactic Dependencies. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 191–199, Vilnius, Lithuania, May 2015. URL https://aclanthology.org/W15-1824.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS 2017)*, pages 5998–6008, Long Beach, USA, January 2017.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: BERT for Finnish. *CoRR*, abs/1912.07076, 2019. URL http://arxiv.org/abs/1912.07076.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, September 2016.

David Yarowsky, Grace Ngai, and Richard Wicentowski. Inducing Multilingual Text Analysis Tools via Robust Projection across Aligned Corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001. URL https://aclanthology.org/H01-1035.

Amir Zeldes and Janet Liu. DISRPT 2021 TASK 2 Results, 2021. URL https://sites.google.com/georgetown.edu/disrpt2021/results#h.gb445xshqmt7. Available at https://sites.google.com/georgetown.edu/disrpt2021/results.

Deniz Zeyrek and Kezban Başıbüyük. TCL - a Lexicon of Turkish Discourse Connectives. In *Proceedings of the First International Workshop on Designing Meaning Representations*, pages 73–81, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3308. URL https://aclanthology.org/W19-3308.

Deniz Zeyrek, Işın Demirşahin, and Ayışığı B. Sevdik Çallı. *Turkish Discourse Bank: Porting a discourse annotation style to a morphologically rich language*, volume 4. 2013. doi: 10.5087/dad.2013.208. URL https://doi.org/10.5087/dad.2013.208. 174.

Deniz Zeyrek, Amalia Mendes, and Murathan Kurfali. Multilingual Extension of PDTB-Style Annotation: The Case of TED Multilingual Discourse Bank. In *Language Resources and Evaluation Conference*, 2018.

Deniz Zeyrek, Amalia Mendes, Yulia Grishina, Murathan Kurfali, Samuel Gibbon, and Maciej Ogrodniczuk. TED Multilingual Discourse Bank (TED-MDB): a parallel corpus annotated in the PDTB style. *Language Resources and Evaluation*, pages 1–38, 2019.

Yuping Zhou and Nianwen Xue. The Chinese Discourse TreeBank: A Chinese Corpus Annotated with Discourse Relations. *Lang. Resour. Eval.*, 49(2):397–431, June 2015. ISSN 1574-020X. doi: 10.1007/s10579-014-9290-3. URL https://doi.org/10.1007/s10579-014-9290-3.

黄江海编著. 戴木金, author. 戴木金, and 黄江海. Guan lian ci yu ci dian, 1988. URL http://reference.apabi.com/hku/book.aspx?bi=m.20081027-m300-w001-049.

宋光中. 王起澜, 张宁, 王起澜., 张宁., and 宋光中. *Han yu guan lian ci ci dian*. Fujian renmin chu ban she, Fuzhou, di 1 ban. edition, 1989.