Lightweight Connective Detection Using Gradient Boosting

Mustafa Erolcan Er[1], **Murathan Kurfalı**[2], **Deniz Zeyrek**[1]
[1]Cognitive Science Dept., Graduate School of Informatics, Middle East Technical University
[2]Sensory-Cognitive Interaction Lab, Department of Psychology, Stockholm University
erolcan@metu.edu.tr, murathan.kurfali@su.se, dezeyrek@metu.edu.tr

Abstract

In this work, we introduce a lightweight discourse connective detection system. Employing gradient boosting trained on straightforward, low-complexity features, this proposed approach sidesteps the computational demands of the current approaches that rely on deep neural networks. Considering its simplicity, our approach achieves competitive results while offering significant gains in terms of time even on CPU. Furthermore, the stable performance across two unrelated languages suggests the robustness of our system in the multilingual scenario. The model is designed to support the annotation of discourse relations, particularly in scenarios with limited resources, while minimizing performance loss.

Keywords: Discourse Connectives, Gradient Boosting, linguistically-informed features

1. Introduction

Recent advancements in deep learning have significantly improved state-of-the-art performances in natural language processing (NLP), and discourse parsing is no exception. Yet, despite these performance gains, these models demand high computing resources, which greatly hinders their usability, as many researchers around the world still lack access. Moreover, these models often act as black-box solutions, without providing any linguistic/theoretical insights regarding the task at hand. In our current submission, we present a lightweight detection system for connectives, which are considered as one of the most important building blocks of discourse structure.

Among various approaches to discourse structure, such as RST (Mann and Thompson, 1987) and SDRT (Lascarides and Asher, 2007), PDTB (Prasad et al., 2014) remains the largest annotated dataset (Prasad et al., 2014) involving discourse-level annotations. PDTB adopts a connective-based approach, where connectives are the anchors of discourse relations that hold between two text spans that have an abstract object interpretation, such as propositions or eventualities (Prasad et al., 2014). The challenge lies in distinguishing between connectives that function as discourse connectives (DC) and those that do not, known as non-discourse connective (NDC) usage. Consider examples (1) and (2):

1. He went to Paris for a vacation and visited the famous Eiffel Tower.
2. He speaks English and French.

(from (Başıbüyük and Zeyrek, 2023))

PDTB recognizes the *and* in the first example as a discourse connective whereas, in the second example, it does not, as it simply links two noun phrases. Thus, the first step in the PDTB annotation process is the detection of the connectives with discourse usage in a given text piece. In the current work, we address this issue using a lightweight model that utilizes linguistic features to efficiently identify discourse connectives without the need for specialized hardware, such as GPUs, which are still not available to most researchers worldwide. We train and evaluate our model in two languages, English (PDTB 2.0) and Turkish (Turkish Discourse Bank (TDB) 1.0 (Zeyrek et al., 2013)). The contributions of our work are:

1. We introduce a fast machine-learning model that detects connectives.
2. We show that this model achieves results close to state-of-the-art models.
3. We argue that verb-based features are the most important aspects of our lightweight connective detection model.

The paper is structured as follows. In Section 2, we introduce two lines of research that deal with connective detection and briefly summarize recently developed discourse parsers that are shown to work in Turkish as well as English. Section 3 introduces our method, and Section 4 the experimental setting as well as the data and baselines. In Section 5 we evaluate our model, and finally, in Section 6 we draw some conclusions.

2. Related Work

Reflecting the overall trend in the field, the literature on discourse parsing can be roughly divided into two parts: the body of works before, and after

ISA-20 Proceedings @LREC-COLING-2024, pages 53–59
20 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

the emergence of neural networks (NNs). Before the solutions based on neural networks became the default approach, the methods relied more on traditional approaches such as feature engineering or annotation projection (Wellner and Pustejovsky, 2007; Pitler and Nenkova, 2009; Versley, 2010).

Following the deep learning revolution, led by the increase in the available computing power and the amount of data, NN-based solutions slowly replaced linguistic features, and more black-box approaches have become popular (Hooda and Kosseim, 2017; Kurfalı, 2020; Kutlu et al., 2023). Most prominently, the recent DISRPT 2021 (Zeldes et al., 2021) and 2023 (Braud et al., 2023) shared tasks have received only transformer (Vaswani et al., 2017)-based solutions to a range of languages including English and Turkish (e.g., Gessler et al., 2021; Metheniti et al., 2023; Anuranjana, 2023), with the exception of the TMVM model by Dönicke (2021), which utilized linguistic features derived from syntactic trees. Gessler et al. (2021) also stands out by integrating linguistics features into transformers.

3. Approach

The proposed connective detection model takes raw natural language data as input and determines which tokens are connectives. The task is modeled as a three-way token classification task, where each token can belong to one of three categories:

- *O*: The token is not part of a connective span.
- *B-Conn*: The token marks the beginning of a connective span. It can represent the entire span of the connective, as in single-word examples like *because*, or the first word of a phrasal connective, such as *on* in *on the other hand*.
- I-Conn: The token is the second or a subsequent word in a phrasal connective, e.g., *other* in *on the other hand*.

A computationally cheap and fast explicit connective detection algorithm should use symbolic or traditional ML-based approaches instead of deep learning architectures. At the same time, the features used by ML-based algorithms should be produced by algorithms with a time complexity lower than the inference time complexity of the ML model. For this purpose, we preferred to use gradient boosting to train our model. Gradient boosting is an ensemble method determining the optimal predictive model to enable us to use the decision trees more effectively (Friedman, 2001).

This iterative algorithm starts with a naive prediction (mostly an average line) to capture the target values. In the second iteration, the residual between this prior prediction and the observed targets is calculated and a decision branch is adapted to decrease the sum of residuals. Repeating this process until the sum of residuals is minimized gives us a final decision tree for our classification task. We use the XGBoost (Chen et al., 2015) library to implement gradient boosting on our datasets.

We decided to incorporate three groups of features to our model. The first group involves verb-based features. These are the main features for our model and involve:

- Whether any of the three words before and three words after a candidate token is verb or not.
- Whether the current word is verb or not.
- The token-based distance of the current word to the previous and the following verbs.

The second group of features involves word-based features consisting of features such as the capitalization of words, word length, and a unique ID assigned to each word in the data, all of which can be produced with O(n) time complexity.

The last group of features includes position-based features, by which we could produce in O(n) time complexity, too. These involve the position of the current word in the sentence, also including the length of sentences based on words.

We used the XGBoost library to train our model with gradient boosting. The XGBoost library offers a wide choice of parameters for gradient boosting. Thus, we performed parameter tuning on learning_rate (contributions of each tree to the final model), max_depth (maximum depth of each tree), n_estimators (number of trees generated by the model), max_delta_step (a parameter that is useful for imbalanced datasets by preventing the weights from updating too much) and min_child_weights (a parameter to control the overfitting problem) which we consider to be the most important ones among these parameters. We used the grid search algorithm (Chicco, 2017) to choose the most effective tuning among these three parameters. Grid search systematically runs the different combinations of parameters and uses cross-validation (Stone, 1974) to find the best combination based on the performance. Recognizing the limited size of our dataset, we applied 3-fold cross-validation in our experiments to ensure a balance between model training time and validation robustness.

The dataset suffers from severe imbalance as discourse connectives do not occur as often. To deal with this, we also train our models with the weighted loss. We used inverse frequency weighting to determine the label weights. That is, for each

Model	Learning Rate	Max Depth	N Estimators	Max Delta Step	Min Child Weights
PDTB 2.0	0.2	8	500	4	1
PDTB 2.0 (Weighted)	0.30	8	400	4	1
TDB 1.0	0.15	10	500	4	1
TDB 1.0 (Weighted)	0.15	8	400	4	1

Table 1: Best Parameters for PDTB 2.0 and TDB 1.0 Datasets. Weighted refers to the classifiers trained with the "weighted" loss.

Dataset	B-Conn	I-Conn	O	Connective Proportion(%)
			TDB	
Training	7,044	1,259	385,256	2.11
Development	773	130	45,939	1.93
Test	849	165	45,944	2.16
			PTDB	
Training	23,848	4,499	1,032,851	2.67
Development	953	159	38,656	2.80
Test	1,245	238	54164	2.67

Table 2: The distribution of labels in the datasets. Refer to Section 3 for the label definitions. The last column denotes the proportion of all connectives to the total number of tokens.

i in our dataset, we computed w_i as

$$w_i = \frac{N}{C \cdot n_i}$$

where N is the total number of instances, C is the number of unique classes and n_i is the number of instances belonging to class i.

Weighted loss is a method used in imbalanced data to ensure that minority class data points contributes more to the model. The idea behind weighted loss is to assign a higher weight to the minority class data points while assign a lower weight to the majority class data points when computing the loss. Thanks to this approach, mistakes on the minority class become more "costly" for the model, causing it to pay more attention to correctly classifying instances of the minority class.

The best parameters according to the grid search are provided in Table 1.

4. Experimental setting

4.1. Data

In our experiments, we followed the training, development, and test splits proposed in DISRPT 2021 (Zeldes et al., 2021) to facilitate direct comparison of our models with the state-of-the-art systems evaluated there. The Turkish data in DISRPT is sourced from TDB 1.0 (Zeyrek et al., 2013), while the English data is based on PDTB 2.0 (Prasad et al., 2008). The distribution of the labels in the respective datasets are provided in Table 2. DISRPT data uses these datasets without any pruning. Thus, our models are trained to explicit discourse connectives including discontinuous connectives such as "if .. then", "either .. or", etc. in addition to continuous or single word connectives. Alternative Lexicalizations (AltLex) connectives are also included in these datasets. AltLexes are not connective on their own but can act as connective when combined as multi word expressions.

4.2. Baseline Models

To put our results into perspective, we compare our model's performance against the best-performing systems in DISPRT 2021 and 2023 shared tasks. Additionally, we report the performance of a vanilla BERT model fine-tuned on the training set[1] (Devlin et al., 2018), to represent the current go-to approach for performing this task. We follow the standard token classification procedure using the default parameters and report the average performance across four different runs. The BERT baseline also provides insights into the time efficiency of our model, as that information is not available for the other baselines. It should be noted that all baselines, except for TMVM, are based on deep neural networks.

[1] We used the *bert-base-cased* for English and the BERTurk model (Schweter, 2020) for Turkish.

Model	Precision (%)	Recall (%)	f-score (%)	Inference Time (sec)
DisCut2023 (Metheniti et al., 2023)	95.49	91.89	93.66	–
DiscoDisco (Gessler et al., 2021)	92.93	91.15	92.02	–
Segformers (Bakshi and Sharma, 2021)	89.73	92.61	91.15	–
DisCut (Ezzabady et al., 2021)	93.32	88.67	90.94	–
TMVM (Dönicke, 2021)	85.98	65.54	74.38	–
BERT Baseline	92.63	91.88	92.25	3.13
Our Model	89.10	78.71	83.58	0.02 (1.33*)
Our Model (Weighted)	70.00	86.02	77.19	0.02 (2.03*)

Table 3: Comparison of the Baseline Models and Our Model over PDTB 2.0 Using DISRPT Data Splits. * denotes inference time on CPU for our lightweight model.

Model	Precision (%)	Recall (%)	f-score (%)	Inference Time (sec)
DiscoDisco (Gessler et al., 2021)	93.71	94.53	94.11	–
DisCut2023 (Metheniti et al., 2023)	92.34	93.21	92.77	–
Segformers(Bakshi and Sharma, 2021)	90.42	91.17	90.79	–
DisCut (Ezzabady et al., 2021)	90.55	86.93	88.70	–
TMVM (Dönicke, 2021)	80.00	24.14	37.10	–
BERT Baseline	92.36	92.89	92.62	5.09
Our Model	87.41	71.96	78.94	0.01 (1.17*)
Our Model (Weighted)	82.42	82.33	82.38	0.01 (1.55*)

Table 4: Comparison of the Baseline Models and Our Models over TDB 1.0 Using DISRPT Data Splits. * denotes inference time on CPU for our lightweight model.

5. Results and Discussion

5.1. Results

We evaluated the performance of our model using the official evaluation script of DISRPT 2021.[2] The evaluation criteria are based on exact span matching, meaning that partial detection of phrasal connectives, such as identifying "*because*" within "*That's because*", does not contribute to the overall accuracy. For each language, micro-averaged precision, recall, and F-scores are reported.

The results of our system for English and Turkish are provided in Table 3 and Table 4, respectively. Despite our model's simplicity and reduced complexity, it demonstrates competitive performance when compared against the strong baselines. The best performances achieved in English and Turkish are very close to each other, suggesting that the model is robust across languages with different linguistic characteristics. Moreover, it must be highlighted that our submission outperforms the feature-based baseline, TMVM, in both languages, with the difference in Turkish being almost threefold. We believe that this finding demonstrates the effectiveness of our set of features and further justifies their applicability to different languages.

Switching to weighted loss led to mixed results. In Turkish, the weighted loss increased the overall performance by 3 points; however, in English, it had a negative effect. Yet, in both cases, weighted loss significantly increased the recall of our models as expected. These findings indicate that while the approach increases the model's ability to identify true positive cases, its impact on precision, hence the overall performance, is language-dependent and requires further investigation.

On the other hand, our models achieved inference speeds at least three times faster than the BERT Baseline, despite being run on a CPU, unlike the BERT model which was trained and evaluated on a GPU. When both models are run on a GPU, the difference becomes nearly 250 times. This confirms that our model is indeed computationally less demanding, making it suitable for scenarios with limited computational resources.

5.2. Feature Importance

After training our model, we performed a feature importance test to determine which features made the highest contribution to the detection of DCs in TDB 1.0 and PDTB 2.0. The most important features detected by our best models in two languages are listed in Figure 1, Figure 2.

[2] https://github.com/disrpt/sharedtask2021

Connective	Number of Correct Predictions		Number of Incorrect Predictions		Accuracy (%)
	True Positive (TP)	True Negative (TN)	False Positive (FP)	False Negative (FN)	
and	204	619	21	40	93.10
for	11	403	1	10	97.41
then	11	2	2	3	72.22
Once	0	0	3	1	0
ve (and)	181	477	33	25	91.90
için (for)	90	88	20	2	89.00
Sonra (After)	15	2	4	2	73.92
aksine (contrary to)	0	1	0	2	33.33

Table 5: Error Statistics for Selected Connectives in English (above) and Turkish (below). The top two connectives are the most frequent ones; the bottom two are the most mispredicted that occur at least three times.

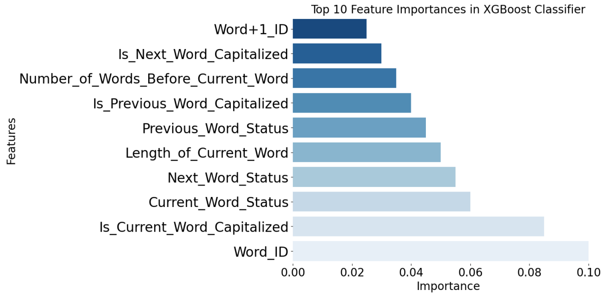

Figure 1: Feature importance in PDTB 2.0 for our best model

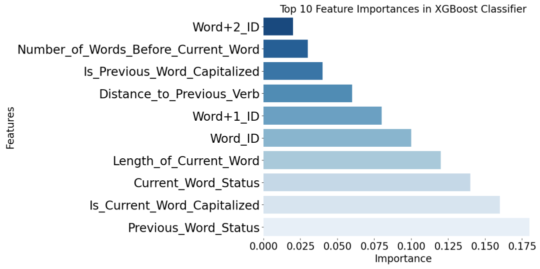

Figure 2: Feature importance in TDB 1.0 for our best model

As seen in the figures, word-based features such as Word ID and Capitalization check are prominent for PDTB. For TDB, the most critical feature is the information on whether the previous word is a verb (Previous_Word_Status). Additionally, while the status of the current word as a verb (Current_Word_Status) significantly contributes to the model for both languages, verb information of the next word for English and the previous word for Turkish stand out. We believe this may be attributed to the differences in word order between Turkish and English.

As shown in (Pitler and Nenkova, 2009), constituent tree-based features such as self category, parent category, sibling category provide very successful results in detecting explicit connectives. However, since annotated trees aligned with raw data are needed to derive these features, deriving these features also has an additional annotation cost. In fact, since the annotation process of a dataset with the PDTB formalism is easier than the constituent tree annotation process, deriving the features to be used for automatic annotation may even cause higher costs than handmade annotation. This shows that our system, in addition to being lightweight compared to deep learning models, is also lightweight compared to classical approaches in terms of producing features effectively and at low cost.

5.3. Error Analysis

In this section, we discuss our model's performance through error analysis. We present the error distribution for selected connectives in Table 5 and discuss some examples. The table highlights the first two connectives as those with the highest occurrence in our dataset, while the last two are identified as the most frequently mispredicted connectives above the specific threshold of 3. For Turkish data, the model tends to over-predict discourse connective (DC) usage over non-discourse connective (NDC) usage while in the PDTB, it is more cautious, often missing instances where connectives serve as DCs.

The examples below are provided to highlight the mistakes of our model. We show the mispredicted tokens by underlining, correctly predicted ones in bold fonts.

Example (4) showcases an unusual case where our model incorrectly identifies a noun in the Turkish dataset, *aklı* ('mind'), as a discourse connective.

4. Laiklik zaten, inançlara saygı duyarak aklı özgürleştirmektir.(False Positive)
   'Secularism already means liberating the mind by respecting beliefs.'

This error is noteworthy because the sentence does indeed contain a connective that expresses a manner relation, specifically through the (intra-sentential) suffixal connective -arak attached to the verb preceding *aklı*. Yet, such suffixal connectives are later added to the TDB in its 1.2 version (Zeyrek and Er, 2022) and are missing in the DISRPT training data. We have spotted several more cases exhibiting the same behavior which suggests that our model is generalizing to the connectives that are not seen in its training data.

Examples (5) and (6) illustrate one of the most common mistakes of our model, both in Turkish and English datasets. In Turkish, it includes a phrasal connective *zaman da* ('when' used with the focus particle); yet, our model only identifies the first part, *zaman* ('when'), missing the focus particle, *da*. In English, the system only recognizes *for*, missing the rest of the connective. Due to the strict evaluation strategy that requires an exact span match, this prediction is classified as misprediction. Overall, the phrasal connectives are particularly challenging.

5. Uygun düştüğü sanıldığı **zaman** da hemen birbirlerinin üzerinden kayıp gideceklerdi. (False Negative)
   '**When** people thought [it] fits, they would immediately slip over each other'.

6. **For** instance, Gannett Co. posted an 11% gain in net income, as total ad pages dropped at USA Today, but advertising revenue rose because of a higher circulation rate base and increased rates. (False Negative)

6. Conclusion and Further Studies

In this study, we introduced a lightweight, gradient-boosting-based system for detecting discourse connectives, achieving competitive performance with significantly faster inference speeds compared to deep learning-based alternatives. Our approach demonstrated robustness across English and Turkish, indicating its utility in multilingual settings and scenarios with limited computational resources. Thanks to the speed and accuracy of our system, our model can be used to mine large amounts of data that can be used to facilitate the development of new discourse-annotated corpora or as the training data of discourse-focused language models.

7. Bibliographical References

Kaveri Anuranjana. 2023. Discoflan: Instruction fine-tuning and refined text generation for discourse relation label classification. In *Proceedings of the 3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 22–28.

Sahil Bakshi and Dipti Misra Sharma. 2021. A transformer based approach towards identification of discourse unit segments and connectives. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 13–21.

Kezban Başıbüyük and Deniz Zeyrek. 2023. Usage disambiguation of turkish discourse connectives. *Language Resources and Evaluation*, 57(1):223–256.

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

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Davide Chicco. 2017. Ten quick tips for machine learning in computational biology. *BioData mining*, 10(1):35.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tillmann Dönicke. 2021. Delexicalised multilingual discourse segmentation for disrpt 2021 and tense, mood, voice and modality tagging for 11 languages. In *Proceedings of the 2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 33–45.

Morteza Kamaladdini Ezzabady, Philippe Muller, and Chloé Braud. 2021. Multi-lingual discourse segmentation and connective identification: Melodi at disrpt2021. In *2nd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2021)*, pages 22–32. ACL: Association for Computational Linguistics.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Luke Gessler, Shabnam Behzad, Yang Janet Liu, Siyao Peng, Yilun Zhu, and Amir Zeldes. 2021. Discodisco at the disrpt2021 shared task: A system for discourse segmentation, classification, and connective detection. *arXiv preprint arXiv:2109.09777*.

Sohail Hooda and Leila Kosseim. 2017. Argument labeling of explicit discourse relations using lstm neural networks. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 309–315.

Murathan Kurfalı. 2020. Labeling explicit discourse relations using pre-trained language models. In *Text, Speech, and Dialogue: 23rd International Conference, TSD 2020, Brno, Czech Republic, September 8–11, 2020, Proceedings 23*, pages 79–86. Springer.

Ferhat Kutlu, Deniz Zeyrek, and Murathan Kurfalı. 2023. Toward a shallow discourse parser for turkish. *Natural Language Engineering*, pages 1–26.

Alex Lascarides and Nicholas Asher. 2007. Segmented discourse representation theory: Dynamic semantics with discourse structure. In *Computing meaning*, pages 87–124. Springer.

William C Mann and Sandra A Thompson. 1987. *Rhetorical structure theory: A theory of text organization*. University of Southern California, Information Sciences Institute Los Angeles.

Eleni Metheniti, Chloé Braud, Philippe Muller, and Laura Rivière. 2023. Discut and discret: Melodi at disrpt 2023. In *3rd Shared Task on Discourse Relation Parsing and Treebanking (DISRPT 2023)*, pages 29–42. ACL: Association for Computational Linguistics.

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

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. 2008. The penn discourse treebank 2.0. In *LREC*.

Rashmi Prasad, Bonnie Webber, and Aravind Joshi. 2014. Reflections on the penn discourse treebank, comparable corpora, and complementary annotation. *Computational Linguistics*, 40(4):921–950.

Stefan Schweter. 2020. Berturk-bert models for turkish. *Zenodo*, 2020:3770924.

Mervyn Stone. 1974. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yannick Versley. 2010. Discovery of ambiguous and unambiguous discourse connectives via annotation projection. In *Proceedings of Workshop on Annotation and Exploitation of Parallel Corpora (AEPC)*, pages 83–82.

Ben Wellner and James Pustejovsky. 2007. Automatically identifying the arguments of discourse connectives. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 92–101.

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

Deniz Zeyrek, Işın Demirşahin, and Ayışığı B Sevdik Çallı. 2013. Turkish discourse bank: Porting a discourse annotation style to a morphologically rich language. *Dialogue & Discourse*, 4(2):174–184.

Deniz Zeyrek and Mustafa Erolcan Er. 2022. A description of turkish discourse bank 1.2 and an examination of common dependencies in turkish discourse. *arXiv preprint arXiv:2207.05008*.