Unsupervised Approaches to the PDTB

Cutter Dalton
University of Colorado Boulder
Cutter.Dalton@colorado.edu

Marie Grace
University of Colorado Boulder
Marie.Grace@colorado.edu

Xajavion Seabrum
University of Colorado Boulder
Xajavion.Seabrum@colorado.edu

Weitung Liao
University of Colorado Boulder
Weitung.Liao@colorado.edu

Abstract

Currently, there is a dearth of research involving unsupervised approaches to discerning sentence and clausal relations in discourse. This research is necessary given the recent advances in state-of-the-art approaches to word and sentence embeddings through bidirectional transformers. This paper focuses on the efficacy of using sentence embeddings encoded through BERT in common unsupervised approaches such as DBSCAN and K-Means to predict explicit discourse relations within the Penn Discourse TreeBank (PDTB) Prasad et al. (2008). Further, this paper explains possible confounding variables that may be hindering supervised learning approaches for predicting sentence discourse relations. While the implementation of DBSCAN methods were limited due to the lack of a provided cluster label as well as inherent information loss, our K-Means approach proved to be more fruitful in producing analyzable clusters due to the ability of providing the number of clusters desired, comparing them to PDTB secondary tags, as well as its ability to scale well with larger datasets. The clusters provided by our K-Means approach demonstrates how sentence embeddings pairs can be mislabled due to their proximity to other pairs belonging to different discourse relations. While some clusters were made up of mostly one relation type, others were split with some majority, and a few had shown no true plurality.

1 Introduction

Recent advances in bidirectional transformers like Google's BERT have led to the implementation of sentence embeddings in the field of natural language processing. While sentence embeddings are intuitively believed to have potential in the field of computational discourse (specifically, sentence relation prediction), their usage in supervised and neural approaches has been fairly underwhelming. (Kishimoto et al., 2020; Huber et al., 2020).

In conjunction with these findings is the extant review of established, popular sentence relation paradigms such as the Penn Discourse TreeBank (Prasad et al., 2008). While PDTB utilizes a rigid and highly specified taxonomy, there are still cases where sentence relation labels are not easily distributed to each pair. Prasad et al. (2014) discuss that a significant proportion (444) of explicit relations had to be reduced to their first level of analysis due to automated backoff. This suggests that there may be some points of data that cannot be easily deliberated between humans, or that the prescribed labels require some form of reexamination.

In this paper, we investigate what is going on "underneath the hood" with sentence relation prediction involving sentence embeddings. This work provides efficient, unsupervised methods for discerning sentence relations within a given discourse. Unsupervised machine learning is relatively underutilized in the field of computational discourse, but it could be especially useful for predicting the discourse relations of sentence embeddings, and understanding why state-of-the-art supervised approaches underperform.

We therefore conduct our experiment on unsupervised methods through different models of KMeans and DBSCAN approaches. We will compare the clusters generated by these unsupervised methods to the gold standard PDTB clusters in order to observe each model's ability to group discourse relations, and also see where potential confusion may be occurring in supervised models.

2 Literature Review

2.1 PDTB Discourse Relations

The release of the PDTB 2.0 in 2008 propagated advancements in the NLP tasks of text summarization, question-answering, and natural language generation. PDTB is significant in how it approached sentence relations within a theory-neutral approach by only examining sentence pairs, and not attempting to establish a hierarchy like that of Rhetorical Structure Theory. This purportedly allows for more flexible research schemes. Further, PDTB provides a lexically based approach to sentence relations, with emphasis on discourse connectives whenever present. These relations are specifically referred to as explicit relations, and make up 18,549 pairs within the 1 million word corpus (Prasad et al., 2008).

This paper will observe these eighteen thousand pairs in its attempt to understand where explicit sentence relation prediction tasks are going wrong with sentence embeddings. In doing so, we hope to facilitate the aforementioned downstream NLP tasks and further research in the field as a whole.

2.2 Unsupervised Relation Recognition

Research on unsupervised approaches to Discourse relation recognition is lacking as a whole. Previous works in the field of computational discourse promising unsupervised approaches to predict sentence relations have not been completely unsupervised. Others have become outdated by recent advances in deep learning. Even with deep learning models such as Huber et al. (2020) and Kishimoto et al. (2020), relation prediction is ineffective or underwhelming. None of the above have attempted to explain where sentence embeddings may go awry for supervised learning.

Sileo et al. (2019) and Nishida and Nakayama (2018) have shown promise in the approach of bidirectional LSTM's to predict discourse relations, but their approaches still employ stochastic gradient descent – reducing their claim of being wholly unsupervised approaches. Despite this caveat, Sileo et al. (2019) demonstrate an effective method to encode sentences, and demonstrate success with predicting the discourse connectives occurring within a sentence which has partly inspired this work.

Early implementations of unsupervised learning such as Marcu and Echihabi (2002) suggest vastly improved results when utilizing large amounts of data. Most importantly, this work establishes the necessity of utilizing unsupervised learning to evaluate discourse relations in the first place. While there is a long standing debate in which theories to uphold and use in the field of discourse, Marcu and Echihabi (2002) also report that there are significant patterns within text above the sentence level which can be viewed and verified by a computer. Although their experiment was conducted utilizing Rhetorical Structure Theory, we believe the same intuitions can be upheld with PDTB.

2.3 Methods for Producing Sentence Representations for Discourse Relations

One of the ways Sileao et al.'s work inspired this paper was through various approaches provided to generate sentence embedding representations. One of which was utilized in our model construction which ended up being the clearest to interpret, and was henceforth used in our analysis. The representation will be discussed more in depth in Section 4.3.

Other important previous findings suggest the potential of embedding the connective in addition to the arguments for predicting sentence relations. In contrast to the findings in Huber et al. (2020), Son and Schwartz (2021) provide a comprehensive and "continuous" approach to sentence relations. In this case, the discourse connective (when extant) would be included in $Arg2$ of the pair of embeddings. This method would not be directly utilized in this experiment, but it did provide grounds to incorporate the connective within the argument embeddings.

3 Data

The data for this paper comes from PDTB-2.0 (Prasad et al., 2008). This paper focuses on the sentences within the PDTB-2.0 dataset that are labeled as $Explicit$, which amounts to 18,459 sentences to examine. Furthermore, this paper focuses only on these three features: $Arg1$ text, $Arg2$ text, and the $Connective$ phrase. These features will serve as the basis that this paper uses when exploring unsupervised clustering methods. The clustering methods will be compared to the 16 PDTB secondary labels.

4 Models

4.1 Unsupervised Models

K-Means and DBSCAN were the two unsupervised approaches for this task. The results of each are documented in Section 5.

4.2 Generating the Embeddings

SentenceBERT from the SentenceTransformer library in Python (Reimers and Gurevych, 2019) was used to formulate the embeddings for the sentences in the PDTB. The embedding techniques by which SentenceBERT was fine-tuned on and were considered for this analysis are: (1) Semantic search embeddings (called `msmarco-distilbert-dot-v5` in the SentenceEncoder library) fine-tuned on a semantic task. Ultimately, the result was less than optimal as this type of embedding caused only one cluster to be generated which is unsuitable for analysis, and (2) SentenceBERT, fine-tuned on QA task (called `multi-qa-MiniLM-L6-cos-v1` in the SentenceEncoder Library)produced the most interpretable embeddings for unsupervised clustering and is the embedding style of choice for the rest of this paper.

4.3 Representation Schemes

Let $[a; b; ...]$ represent vector concatenation containing elements a, b, etc. and let $a \cdot b$ represent the dot product between two arguments. Furthermore, let E_1 be the SentenceBERT embedding for $Arg1$, E_2 be the SentenceBERT embedding for $Arg2$, and let C be the SentenceBERT embedding for the connective. The authors of this paper explored the following three vector representations to pass into the unsupervised techniques:

- Representation 1: $[E_1; C; E_2]$, resulting in an embedding vector length of 1,152.
- Representation 2: $[E_1; C; E_2; |E_1 - E_2|; E_1 \cdot E_2]$, as inspired by (Sileo et al., 2019). This representation results in an embedding vector with a length of 1,537.
- Representation 3: $[E_1 \cdot E_2]$, resulting in a scalar value.

Of the three representation schemes above, representation 2 proved to be the most interpretable and is the representation that this paper uses for its analyses below.

5 Unsupervised Results

5.1 DBSCAN

Given a set of points in some space, density-based spatial clustering of applications with noise (DBSCAN) groups together points that are closely packed together, marking as outliers points that lie alone in low-density regions whose nearest neighbors are too far away. DBSCAN does not require a preselected number of clusters and assumes densities across clusters are the same (Ester et al., 1996).

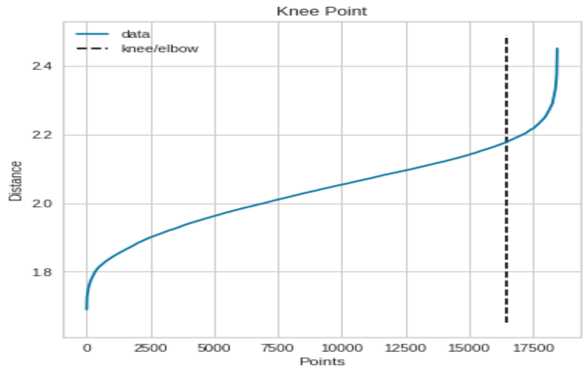

Figure 1: $kneedle$ graph of KNN approach to find optimal distance/epsilon of 2.18 using k = $\sqrt{N}$.

DBSCAN takes the hyperparameters *epsilon* for distance and the *minimum samples* that could create a cluster. We used a systematic nearest neighbor approach to tune epsilon. Epsilon is proportional to the expected number of neighbors, so we can find an appropriate value of epsilon. Two approaches were used to find the expected number of nearest neighbors. The first approach is a k-nearest neighbors approach using $\sqrt{N}$ for N samples. The graph is shown in Figure 1. The other k value tried is $N/16$ as we naively expect 16 groups as the PDTB has 16 secondary labels. *kneedle* (Satopaa et al., 2011) was used to ultimately determine the optimal epsilon value. Both approaches found an epsilon value greater than 2, which is too large. Large epsilons tend to include more points within a cluster and consequentially it considered nearly every point to be in one big cluster with some uncluttered points labeled as noise.

This follows the main disadvantage of DBSCAN, that it does not perform as well on higher dimensional data because the Euclidean distance does not transfer as well as a metric, and as such, it by itself was not useful for exploring clusters.

5.1.1 DBSCAN with T-SNE

Considering the main disadvantage of DBSCAN, scikit-learn's (Pedregosa et al., 2011) T-SNE (t-distributed stochastic neighbor embedding), was used in conjunction to reduce the number of features. T-SNE models each high-dimensional object by a two-dimensional point, where similar objects

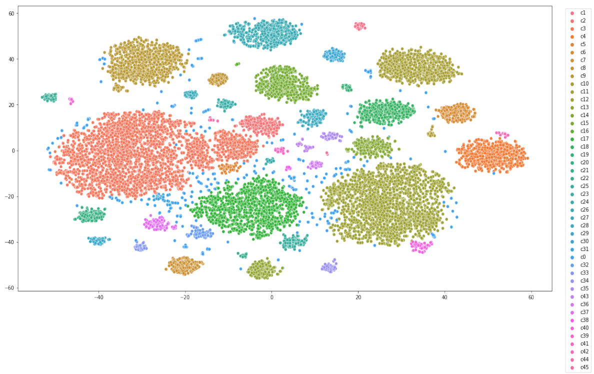

Figure 2: Visualizations of DBSCAN with T-SNE with *eps* value of 1.8 and *min samples* value of 20. Forty-four clusters are found, with cluster 0 containing the points considered as noise

are modeled by nearby points and dissimilar objects are modeled by distant points with high probability. T-SNE is better in this case than a traditional PCA feature reduction approach because it is non-linear and retains the local structure of data which translates to preserving the clusters, whereas PCA attempts a linear approach to preserve the global structure of the data. The two-dimensionality allows the Euclidean distance metric of DBSCAN to excel, as well as provide visualizations that provide insight into tuning the hyperparameters. One potential disadvantage is that T-SNE also requires a hyperparameter perplexity. The default recommended perplexity of 30 was used, but it is easy to lose information when reducing higher dimensional data to two dimensions. Additionally, a distance-based metric then may not capture exactly the trends of the original data. The optimal *epsilon* was found to be 1.8 and the optimal *min-samples* was found to be 20 for the data. The number of clusters is found to be 44 through this approach. A visualization of these clusters can be found in Figure 2.

5.1.2 DBSCAN Limitations

DBSCAN provides a promising approach to unsupervised clustering for discourse connectives. Further work should be done on more thoroughly tuning the hyperparameters. For the purposes of this project, K-Means proved to be more fruitful as it does not induce information loss as T-SNE does and the number clusters can be more directly com-

			K-means Cluster Label								
PDTB Label			0	1	2	3	4	5	6	7	8
Primary	*Secondary*	# sent.	**3100**	**3803**	**989**	**866**	**2870**	**2138**	**2466**	**858**	**1369**
	asynch.	**2022**	0.2	0.1	23.2	2.7	0.8	—	70.5	—	0.1
temporal	synch.	**1413**	—	—	56.5	59.8	10.5	—	1.3	—	—
	P: temp.	**5**	—	—	0.5	—	—	—	—	—	—
	cause	**1818**	2.1	—	2.0	33.5	1.5	—	22.1	99.5	0.1
	condition	**1356**	0.1	—	15.0	1.8	—	—	0.9	—	85.1
contingency	prag. cause	**8**	—	—	—	0.2	0.1	—	—	0.2	—
	prag. cond.	**67**	0.1	—	1.3	—	—	—	—	0.2	3.7
	P: conting.	**1**	—	—	0.1	—	—	—	—	—	—
	concession	**1201**	—	15.4	1.0	0.7	18.8	—	0.6	—	3.2
	contrast	**3844**	0.7	73.1	0.2	0.1	35.5	—	0.6	—	0.4
comparison	prag. conc.	**11**	—	0.1	—	—	0.2	—	—	—	0.1
	prag. contr.	**18**	—	0.4	—	—	—	—	—	—	0.1
	P: comp.	**397**	—	8.2	—	0.2	2.7	—	0.2	—	0.1
	alternative	**351**	2.9	0.1	0.2	0.5	4.0	0.3	1.7	—	6.5
	conjunction	**5212**	87.0	2.2	—	0.1	9.9	98.8	1.4	—	—
	exception	**14**	—	0.3	—	—	—	—	0.1	—	—
expansion	instantiation	**302**	—	—	—	—	10.5	—	—	—	—
	list	**240**	6.8	—	—	—	0.4	0.6	0.3	—	—
	restatement	**155**	0.1	0.1	—	0.1	4.8	0.2	0.2	—	0.1
	P: expans.	**24**	0.1	0.1	—	0.2	0.2	—	0.1	—	0.7

Table 1: The above shows what percent of a K-means cluster is comprised of a certain PDTB secondary label. For example, 87.0% of K-Means cluster 0 contains sentences that were labelled as conjunction. Labels with P: were sentences that had a primary label but no secondary label. Cells with a dash (—) indicate that that cell's value is 0 or rounds to 0.

pared with the labels used by PDTB.

5.2 K-Means

K-Means is an unsupervised clustering approach that groups the data into *k* clusters where each observation is a member of the cluster with the nearest mean. The advantages of using K-Means is that it scales well to larger data sets and is relatively simple to implement. The disadvantages include that it does not handle noise well, and *k* must be chosen manually. Up to 30 clusters were scanned through in searching for the optimal number of clusters, and *kneedle* was used to identify the optimal *k*. The optimal *k* value was determined to be 9. K-means was then run on the data, clustering the sentences' representations into 9 distinct groups. The results of the groupings can be seen in Table 1. The number of sentences present within either the K-means cluster or the PDTB secondary label is presented within the table along with column percent values representing within each K-means cluster what percent of that cluster is comprised of different PDTB secondary labels. All clusters, except for cluster 4, have one of the secondary relations as the majority class.

For cluster 0, *conjunction* has a strong presence along with *list*. This is likely explained by sentences using the word "and". Cluster 1 has *contrast* as its majority class with *concession* being the second most represented class in the cluster. This mixture is likely influenced by sentences in the cluster using connectives like "except" and "however". Clusters 2 and 3 have *synchronous* as the majority label: cluster 2 is more temporally focused (as *asynchronous* relations are also represented) and cluster 3 incorporates sentences labelled as *cause* (which can be explained with the connectives "as" and "as a result").

Clusters 5 through 8 have strong majority classes (*conjunction*, *asynchronous*, *cause*, and *condition* respectively). Note however that *concession*, *alternative*, *instantiation*, *list* and *restatement* are not a majority in any of the k-means clusters. Possible reasons why are discussed in Section 6. Additionally, *pragmatic cause*, *pragmatic condition*, *pragmatic concession*, *pragmatic contrast*, and *exception* also aren't represented in the clusters, however, this is more than likely due to their severely limited representation in the dataset as a whole.

5.2.1 Potential Problematic Sentences (PPS)

Because there seems to be many disparate secondary labels in cluster 4, the sentences that are in this cluster will be considered as "potential problematic sentences" (PPS). To see what is going on with cluster 4, K-means was run again just on the sentences in it to see if further distinctions can be detected in an unsupervised manner. The optimal number of groups suggested was 5 and the results of these groupings can be seen in table 2. As can be seen, all K-means clusters, except for cluster 4.2, have a secondary class as its majority class, or, as in the case of cluster 4.4, two classes as the plurality.

			PPS Clusters				
PDTB Label			4.0	4.1	4.2	4.3	4.4
Primary	*Secondary*	# sent.	**317**	**781**	**926**	**193**	**653**
temporal	asynch.	**24**	—	—	2.6	—	—
	synch.	**302**	—	25.9	0.6	48.7	—
contingency	cause	**43**	—	—	4.6	—	—
	condition	**1**	—	—	0.1	—	—
	prag. cause	**4**	—	—	0.4	—	—
comparison	concession	**540**	—	10.9	15.6	0.5	47.5
	contrast	**1018**	—	52.5	29.3	14.5	47.3
	prag. conc.	**5**	—	—	0.5	—	—
	P: comp.	**78**	—	2.7	2.8	—	4.7
	alternative	**116**	—	—	12.5	—	—
	conjunction	**284**	0.3	6.9	17.2	36.3	—
	exception	**1**	—	—	—	—	0.2
expansion	instantiation	**301**	94.0	—	0.3	—	—
	list	**11**	—	1.2	0.2	—	—
	restatement	**137**	5.7	—	12.6	—	0.3
	P: expans.	**5**	—	—	0.5	—	—

Table 2: The above shows the clusters generated for the Potential Problematic Sentences. For example, 94.0% of K-Means cluster 4.0 contains sentences that were labelled as instantiation. Labels with P: were sentences that had a primary label but no secondary label. Cells with a dash (—) indicate that that cell's value is 0 or rounds to 0.

6 Probable Implications and Future Direction

Why do some clusters struggle to have a majority class? Furthermore, for the clusters that do have a majority class, why do some clusters have the same majority class?

One possibility is that with the representations as given, the signal from the connective or from the arguments is not strong enough for an algorithm to detect a difference. To visualize this, Figure 3 shows that the representation does not cleanly deliniate with respect to the PDTB labels.

This could have implications for future research into PDTB-2.0, especially for neural net

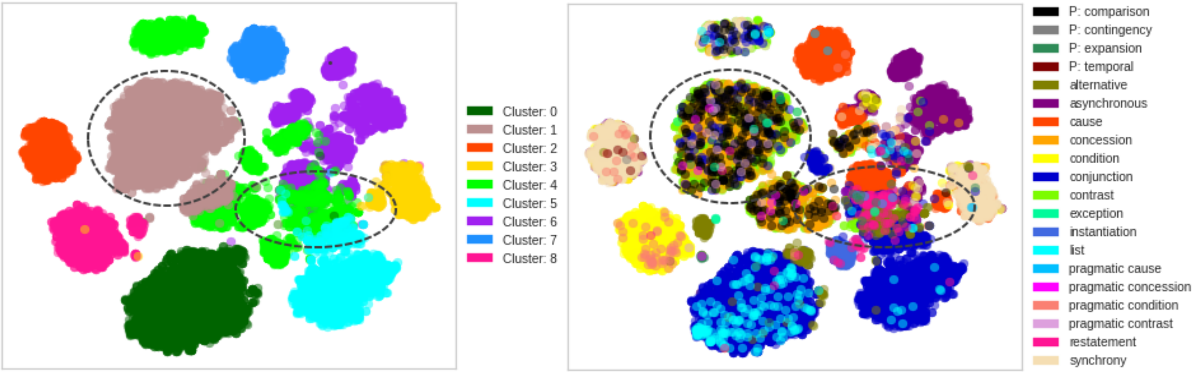

Figure 3: K-Means (left) and PDTB groupings (right) projected down to 2-D space with colors representing different labels. The PPS sentences are marked with the wider oval. Another potential group to investigate (roughly corresponding to K-means cluster 1) is marked with a more circular oval.

approaches to these data. So far, the best results from a neural perspective on the data come from (Kishimoto et al., 2020); however, the model accuracy for certain labels is only just above 55%, with the rest being below 50%.

Because there does not seem to be a meaningful enough distinction for some of the sentences in PDTB-2.0, it might be advantageous to consider a curriculum learning approach. To expound, neural model performance might improve if neural networks were trained on easier, more prototypical (non-PPS) sentences for each of the secondary labels first and then trained on harder (PPS sentences and the other group circled in Figure 3) sentences.

7 Limitations

There were a few limiting factors regarding the models and the project as a whole. In general, time constraints limited what could fully be done with the analysis and implementation of the project. In order to provide a sufficient examination of explicit sentence embeddings, implicit relations were eschewed for the time being. Furthermore, complementary analysis of assessing the quality of the K-means clusters, as described in (Amigó et al., 2009)'s approach, was not performed and would have likely enriched the discussion presented in this paper.

In addition, the results delivered by these models are not the final say in unsupervised approaches, and although the findings do suggest why sentence embeddings may be less effective in predicting discourse relations, more research should be conducted utilizing other methods of unsupervised clustering to disprove or corroborate the results found here. To expand on that, only two main algorithms are considered, and neither is perfect for this task. K-Means can have difficulty clustering data of varying size and density while DBSCAN has trouble on larger datasets, but plenty more algorithms exist that could have strong performance for this task. Other models like mean-shift clustering, affinity propagation, and hierarchical clustering were unable to be implemented and evaluated due to the length of runtime and the memory required.

Furthermore, this analysis is limited to just the data as presented in PDTB-2.0. Any changes that have happened with PDTB-3.0 (such as sentences having more than one label) are not present in this analysis. Furthermore, while this paper analyzed $Explicit$ relationships, $Implicit$ and other relationships are not taken into consideration.

8 Conclusion

While unsupervised methods may face significant challenges in assigning accurate relation labels, they can provide pertinent information regarding confounding variables or underlying issues involved with supervised machine learning. As of now, unsupervised machine learning remains underutilized in predicting discourse relations, especially in cases where sentence embeddings are integral to the model.

Some sentence embedding pairs belonging to different relation labels will be grouped within the same cluster, which may cause them to be given the same label in a supervised setting. More work should be conducted in investigating what features should be emphasized in the embeddings of each relation label to assist in further distinguishing them from each other.

References

Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. 2009. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Laurine Huber, Chaker Memmadi, Mathilde Dargnat, and Yannick Toussaint. 2020. Do sentence embeddings capture discourse properties of sentences from scientific abstracts? In *CODI 2020-EMNLP 1st Workshop on Computational Approaches to Discourse*.

Yudai Kishimoto, Yugo Murawaki, and Sadao Kurohashi. 2020. Adapting bert to implicit discourse relation classification with a focus on discourse connectives. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1152–1158.

Daniel Marcu and Abdessamad Echihabi. 2002. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th annual meeting of the association for computational linguistics*, pages 368–375.

Noriki Nishida and Hideki Nakayama. 2018. Coherence modeling improves implicit discourse relation recognition. In *Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue*, pages 344–349.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The penn discourse treebank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. 2011. Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171.

Damien Sileo, Tim Van-De-Cruys, Camille Pradel, and Philippe Muller. 2019. Mining discourse markers for unsupervised sentence representation learning. *arXiv preprint arXiv:1903.11850*.

Youngseo Son and H Andrew Schwartz. 2021. Discourse relation embeddings: Representing the relations between discourse segments in social media. *arXiv preprint arXiv:2105.01306*.

A Hardware Specifications and Code

The models described in this paper were run in Python 3.7.13 in Google Colab, using at most 8 GB of RAM. The code that the authors used can be found here: `https://github.com/xjseabrum/comp_disc_S22_proj`

B Supplemental Material

This section is devoted to representing the clustering data with respect to the primary label and with respect to the tertiary label. These tables can be found on the page that follows.

		K-means Cluster Label								
		0	1	2	3	4	5	6	7	8
Primary PDTB Label	# sent.	**3100**	**3803**	**989**	**866**	**2870**	**2138**	**2466**	**858**	**1369**
temporal	**3440**	0.2	0.1	80.2	62.5	11.4	—	71.8	—	0.1
contingency	**3250**	2.3	—	18.4	35.6	1.7	—	23.0	100.0	88.8
comparison	**5471**	0.7	97.2	1.2	1.0	57.2	—	1.4	—	3.8
expansion	**6298**	96.8	2.7	0.2	0.9	29.8	99.9	3.8	—	7.3

Table 3: The above shows what percent of a K-means cluster is comprised of a certain PDTB primary label. For example, 96.8% of K-Means cluster 0 contains sentences that were labelled as expansion. Cells with a dash (—) indicate that that cell's value is 0 or rounds to 0.

PDTB Label				***K-means Cluster Label***								
				0	1	2	3	4	5	6	7	8
Primary	*Secondary*	*Tertiary*	# sent.	**3100**	**3803**	**989**	**866**	**2870**	**2138**	**2466**	**858**	**1369**
temporal	asynch.	precedence	**950**	0.2	0.1	0.4	—	0.6	—	37.4	—	—
		succession	**1069**	—	—	22.8	2.7	0.2	—	33.0	—	0.1
		S: asynch.	**3**	—	—	—	—	—	—	0.1	—	—
	synch.	S: synch.	**1413**	—	—	56.5	59.8	10.5	—	1.3	—	—
	P: temp.	P: temp.	**5**	—	—	0.5	—	—	—	—	—	—
contingency	cause	reason	**1201**	—	—	2.0	24.5	—	—	4.6	99.5	—
		result	**617**	2.1	—	—	9.0	1.5	—	17.4	—	0.1
	condition	factual past	**9**	—	—	—	—	—	—	—	—	0.7
		factual present	**90**	—	—	0.4	0.2	—	—	—	—	6.1
		general	**327**	—	—	13.3	0.7	—	—	0.2	—	13.4
		hypothetical	**751**	0.1	—	1.2	0.9	—	—	0.7	—	51.9
		unreal past	**54**	—	—	—	—	—	—	—	—	3.9
		unreal present	**123**	—	—	—	—	—	—	—	—	8.9
		S: cond.	**2**	—	—	—	—	—	—	—	—	0.1
	prag. cause	justification	**8**	—	—	—	0.2	0.1	—	—	0.2	—
	prag. cond.	implicit assertion	**46**	0.1	—	1.2	—	—	—	—	0.2	2.2
		relevance	**21**	—	—	0.1	—	—	—	—	—	1.5
	P: conting.	P: conting.	**1**	—	—	0.1	—	—	—	—	—	—
comparison	concession	contra-expectation	**798**	—	14.9	0.1	0.2	7.4	—	0.5	—	0.4
		expectation	**386**	—	0.4	0.9	0.5	11.1	—	0.1	—	2.7
		S: conc.	**17**	—	0.2	—	—	0.3	—	—	—	0.1
	contrast	juxtaposition	**1140**	0.4	19.1	—	—	13.4	—	0.4	—	0.3
		opposition	**347**	0.2	5.4	0.1	0.1	4.6	—	—	—	—
		S: contr.	**2357**	0.1	48.5	0.1	—	17.5	—	0.2	—	0.1
	prag. conc.	S: prag. conc.	**11**	—	0.1	—	—	0.2	—	—	—	0.1
	prag. contr.	S: prag. contr.	**18**	—	0.4	—	—	—	—	—	—	0.1
	P: comp.	P: comp.	**397**	—	8.2	—	0.2	2.7	—	0.2	—	0.1
expansion	alternative	chosen alternative	**115**	—	0.1	—	0.2	2.9	—	1.1	—	—
		conjunctive	**47**	1.3	—	—	—	0.1	—	—	—	—
		disjunctive	**143**	0.4	—	0.2	0.2	0.9	0.2	0.3	—	6.5
		S: alternative	**46**	1.1	—	—	—	0.1	0.1	0.2	—	—
	conjunction	S: conjunction	**5212**	87.0	2.2	—	0.1	9.9	98.8	1.4	—	—
	exception	S: exception	**14**	—	0.3	—	—	—	—	0.1	—	—
	instantiation	S: instantiation	**302**	—	—	—	—	10.5	—	—	—	—
	list	S: list	**240**	6.8	—	—	—	0.4	0.6	0.3	—	—
	restatement	equivalence	**12**	—	—	—	—	0.4	—	—	—	—
		generalization	**16**	—	—	—	—	0.4	—	0.1	—	—
		specification	**109**	—	0.1	—	0.1	3.4	0.2	0.1	—	0.1
		S: restatement	**18**	—	—	—	—	0.6	—	—	—	—
	P: expans.	P: expans.	**24**	0.1	0.1	—	0.2	0.2	—	0.1	—	0.7

Table 4: The above shows what percent of a K-means cluster is comprised of a certain PDTB tertiary label. For example, 87.0% of K-Means cluster 0 contains sentences that were labelled as conjunction. Labels with P: were sentences that had a primary label but no secondary label. Labels with S: were sentences that had a secondary label but no tertiary one. Cells with a dash (—) indicate that that cell's value is 0 or rounds to 0.