# Dependency and Coreference-boosted Multi-Sentence Preference model

**Farhad Mohsin, Inwon Kang, Yuxuan Chen, Jingbo Shang and Lirong Xia**

**Abstract**

Comparative preference classification (CPC) is the task of identifying preferences between entities expressed in text. Multitudes of preferences are expressed in online reviews, forums, and so on, and classifying this gives more information about users and various entities. It is natural that the preference is conveyed across different sentences with multiple (possibly irrelevant) entities mentioned, however, pioneer works only focus on single sentences with exactly one mention of both entities. We consider a dataset with three-entity discussions in addition to two-entity ones. Additionally, we note that 92% of the texts have more than one sentence. We propose to construct a graph spanning multiple sentences to capture the relationship between different entities and then leverage graph neural networks to learn preferences. Specifically, we build multi-sentence graphs using dependency graphs, coreference chains between sentences, and linking of synonyms to build generalized methods. We also consider positional encodings for additional information from multiple mentions of entities. We find that the task becomes more difficult for complex texts, and our new MultiSentPref method achieves a 57% average F1 score.

## 1 Introduction

Comparative preference classification (CPC) is the task of predicting preference between pre-defined entities in natural language. The task consists of first predicting whether there is a preference. If it exists, the next task is to predict the preference between two entities. CPC is an important learning problem in the field of natural language processing (NLP) because of how commonly this problem occurs. Comparative preference learning is essential in search engines, recommender systems, group decision-making systems, among other use cases. Imagine that the comparison is between two colleges, Caltech and Duke, being discussed as admission choices for a first-year student. The preference may be expressed as simply as "Caltech is a better fit for her than Duke." Or it can be accompanied by explanations etc., throughout a few sentences like this: "Caltech is a good school. It has a great atmosphere. But because of her personal preferences, I think she would be happier in Duke". Both are commonplace in natural discussions. Formally, given a comparative text $t$, $k$ entities $c_1, c_2, \ldots, c_k$, where we know that each entity is relevant to the discussion, our goal is to classify the preference relation between each pair of entities $c_i, c_j$. We consider three possible preference relations: $c_i \succ c_j$, $c_i \prec c_j$, and no preference between $c_i, c_j$.

Techniques to solve this problems limited to single sentences with two entities where each entity is mentioned only once. CompSent-19 (Panchenko et al. 2019) is a commonly used benchmark for the CPC task, which has a SOTA graph neural net based solution (Ma et al. 2020) and sentences in this dataset also make this same assumption. But, we have seen that this assumption is unrealistic in many cases. Given this issue, we redefine the CPC task for a more general setting. The target may span through multiple sentences, have multiple mentions of entities, and even consider more than two entities. Recently, Haque et al. (2022) discussed the problem of implicit preferences where one of the entities might not be mentioned, but also focus on single sentences. We consider a different dataset that is more realistic and contains more general cases. We use the College Confidential dataset (Mohsin et al. 2021), which consists of discussions in a forum about different college admission options. Each discussion is a collection of texts, discussing multiple entities. The discussions included in the dataset discuss either two or three entities. The inclusion of three entities already makes the task different from regular CPC. Additionally, we note that $>$90% of the texts in this dataset has more than one sentences. Only $\sim$30% of the texts mention both entities. And in the texts that mention at least one entity, $\sim$69% of them have repeated mention of some entity. In many of these cases, the repeated mentions are not direct but are rather referenced using synonyms or pronouns. This makes coreference resolution for these entities also an important problem.

We present a general method, Dependency and Coreference-boosted Multi-Sentence Preference Model (MultiSentPref), to work with multi-sentence texts with multiple mentions. We create multi-sentence spanning graphs using dependency relations and use graph attention networks (Veličković et al. 2018). The graph edges come from dependency graphs (Chen and Manning 2014; Dozat and Manning 2016) for each sentences. The node features are word embeddings for the words in the sentences. We create additional edges by finding synonymous mentions and coreference chains (Lee, He, and Zettlemoyer 2018) for entities. Figure 1, in the final graph shows both dependence and coreference relations. Finally, we have seen that the position of entities in sentences also

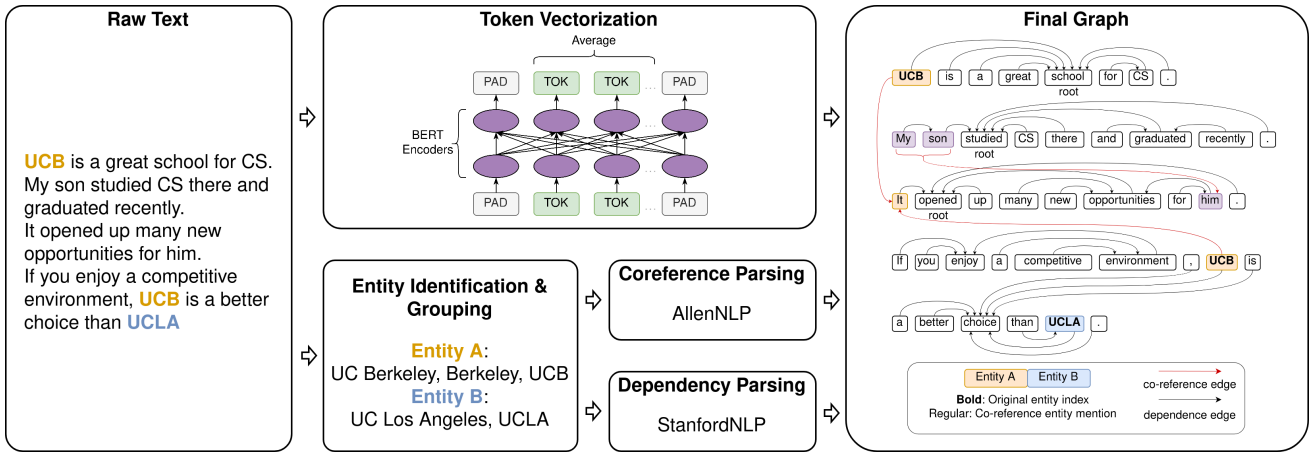


Figure 1: Dependency and coreference graph for a multi-sentence text.

be important features in comparative sentences. We also add positional embeddings to the words to include positional information in the model. Figure 1 gives a high level idea of how the multi-sentence graph is built. We find that the more realistic setting for the CPC task that we address is harder to solve than the single-sentence instances. Existing methods directly applied to the College Confidential dataset does not get high accuracy. However, our model with the added information gets a macro-F1 score of 57% on the three-class classification problem of finding preference between entities.

## 2 Preliminaries

In this section, we discuss the background behind dependency parsing and coreference resolution that builds our multi-sentence spanning graph representation of sentences. We further discuss word/token level embeddings and sentence embeddings, which will be. Finally, we present the basics of the graph attention network, which is the deep graph neural network model that works as the backbone of the MultiSentPref model.

**Dependency parsing:** A dependency tree is a grammatical structure of the sentence represented in a graph form. In previous work, (Ma et al. 2020) used the Stanford Neural Network Dependency Parser (Chen and Manning 2014) to create their edges. However, we found that the performance of the previous benchmark data (CompSent-19) on different implementations of dependency parsers were equivalent, so we decided to used Python's stanza library developed by the same group for computational performance which used a newer technique (Dozat and Manning 2016). In most work in comparing entities, before passing the text to the dependency parser, the text is tokenized by spaces, and the mentions of each entity is replaced with a mask, such as *Entit A* or *Entity B*. This was done to make sure that each entity would be treated as a single token.

**Coreference resolution:** Coreference resolution is the problem of connecting mentions of entities back to their original entities. We use the Stanford CoreNLP implementation for coreference resolution (Clark and Manning 2016b,a).

**BERT embedding:** BERT Devlin et al. (2019) is a pre-trained deep bidirectional transformer model that allows us to generate pre-trained contextual embeddings. In particular, we use bert_uncased_L-12_H-768_A-12 to create the embeddings for our sentences. If a text is greater than 510 tokens, which is BERT's maximum length excluding the padding in the front and back, we truncated the sentence before tokenizing.

**Positional embeddings:** Since we do not have any assumptions about position (or relative position) in text for the entities being compared, we need to manually encode that information. We use the following positional embedding (similar to (Vaswani et al. 2017)): $PE(pos, 2i) = \sin(pos/10000^{(2i/d)})$ and $PE(pos, 2i + 1) = \cos(pos/10000^{(2i/d)})$, where $d$ is the dimension of the positional embedding. We consider $d$ as the length of the BERT embeddings we use for the node feature and add the positional embedding to them.

**Graph Attention Networks:** Graph Attention Networks (GAT) (Veličković et al. 2018) are a special class of graph neural networks that makes use of graph structural information along with regular features. It applies convolution function on neighboring nodes while aided with a self-attention mechanism to automatically find which neighbors are more important to a node for the task at hand. The full GAT model is built using multiple graph attention layers. Abusing notation, we call these individual layers GAT layers. For a full GAT model, multiple individual GAT layers are usually stacked together, followed by more operations depending on the task at hand. In short, getting the output of layer $\ell$ from inputs $H^\ell = \{h_1^\ell, \ldots, h_n^\ell\}$ can be summarized as

$$H^{\ell+1} = GAT(H^\ell, A; \Theta^\ell)$$

where $A$ is the adjacency matrix of the graph and $\Theta^\ell$ includes all the parameters for level $\ell$. Figure 2 gives a high level description of the GAT-based architecture that we use (further described in Section 4).

| | Two-entity texts | | | Three-entity texts | | | | |
|---|---|---|---|---|---|---|---|---|
| Entities mentioned | 0 | 1 | 2 | 0 | 1 | 2 | 3 | Total |
| $A \succ B$ | 54 | 184 | 191 | 33 | 72 | 41 | 23 | 598 |
| $A \prec B$ | 44 | 150 | 162 | 22 | 85 | 48 | 33 | 544 |
| No preference | 463 | 252 | 213 | 401 | 287 | 145 | 61 | 1822 |
| Total | 561 | 586 | 566 | 456 | 444 | 234 | 117 | 2964 |

Table 1: Distribution of label types in the College Confidential dataset

## 3 Exploring the College Confidential Dataset for the CPC task

For the College Confidential dataset (Mohsin et al. 2021), each instance is a single pairwise preference expressed as part of a discussion using natural language. As mentioned in the introduction, each instance may consist of one or more sentences. For simplicity, we call the comparative text for each instance simply as *text*. In each comment, for each pair of entities (say $c, d$), the preference takes one of the following forms $c \succ d, d \succ c, c = d, None$. For this work, we merged the two labels $c = d$ and *No preferences* in accordance with common practice and treated the common label as *None*. We present the distribution of the labels in Table 1.

For a fixed discussion, the compared entities always remain the same. And the dataset contains discussions with either two or three entities being discussed. As we see in Table 1, there are many sentences where not both entities are mentioned. Additionally for three entity-discussions, there are some texts comparing all three entities. Mohsin et al. (2021) break down three entity comparisons into pairwise comparisons, thus each three-entity text leading to three instances. The fact that there can be an additional entity in text which is not one of the two compared entities pose more difficulty. This motivated us to include positional information of the entities being considered.

| **Sentence** | **Label** |
|---|---|
| If Duke is more expensive then go to UCB. Your parents will thank you for saving them money by going there. And you will have more access to jobs in Calif when you graduate. To me its a no brainer. | C >D |
| Cal is notably stronger in the field you want to study, so if you are certain CS is it for you, it may be better to go there. Even full pay parents love a bargain, and will thank you for saving them money! | C >D |
| Daughter graduating from Cal next month. full disclosure. To me, Cal is a no brainer. | C >D |
| You also have to live there and enjoy the people around you. Your education also includes the experiences and people you meet outside of the classroom and labs. Visit both places and make your decision | NA |
| Both are really good schools. You cannot make a mistake going to either place. | C = D |

Table 2: Example comparative texts from a single discussion in the College Confidential dataset

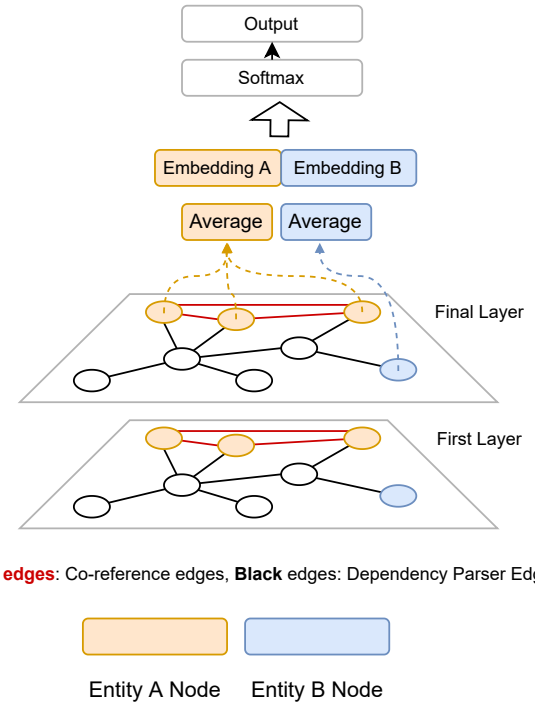


Figure 2: A multi-layer GAT model for comparative text classification on multi-sentence spanning graph representations

We see some representative texts from the dataset in Table 2. As noted before, all of these are multi-sentence texts, which is quite common in the dataset. Many of them mention the same entity more than once. Because of the multi-sentence structure, pronouns are used, making it necessary to use coreference resolution to find all mentions. Most significantly, some of the discussions, and thus some of the texts, in the College Confidential dataset has three entities. In some of these texts, despite there being three entities, only two are compared, whereas in some other all three are compared. The dataset does not present the comparisons as rankings but rather as three pairwise comparisons. We do not differentiate between two-entity and three-entity texts in any of our experiments and treat the pairwise comparison label similarly for both cases in terms of training our models.

## 4 MultiSentPref Model for the CPC task

The MultiSentPref model starts by building a graph representation of a sentence. We use the Stanza dependency parser, which is based on the implementation in (Dozat and Manning 2016). The dependency parser both finds pair of words with dependency between them and also the specific dependency relation. We add an undirected edge between two nodes, whenever there is a dependency from one to other, saving the dependency type as an edge attribute.

To deal with multi-sentence texts, we then connect the disjoint graphs that represent individual sentences. First we connect the root of each sentence to a dummy root node

to get a fully connected graph. In the College Confidential dataset, many texts have multiple mentions of the same entity. To find these additional relations, we use coreference resolution to find chains of coreference mentions within the span of text. We use AllenNLP's coreference analyzer (based on (Lee, He, and Zettlemoyer 2018)) for this task. Using the coreference analyzer, we also added to the entity mentions count that our previous methods did not find, such as cases of pronouns. Sometimes, there are synonymous mention of the same entity (e.g., "University of California, Berkeley" and "UCB" meaning the same entity). We noticed that sometimes the coreference analyzer fails to find these synonymous mentions. So, we use a manually curated list of synonyms for the entities to add such relations.

Once the graph parsings are complete, we encode each 'token' of the text to BERT embeddings and finally add the positional embeddings. When creating these entity embeddings, we use the original text in order to capture any semantics the entity word itself might have had. In cases where the entity is composed of multiple words (e.g., "University of California Berkeley", four words), we take the average of the embeddings of each token in the entity to create a single vector for the entity. After creating the embedding for entities, we mask them with masks such as *entityA* or *entityB*, so that each entity mention is represented by a single node.

The graph neural network model that we apply is a multilayer GAT with the GAT layers stacked on top with a linear layer at the end. Since a single GAT layer indicates relations with one-hop neighbors in terms of the dependency graph, we need multiple layers to properly find multi-hop relations across multi-sentence texts. In our experiments, we find that 20 layers give us the best performance. The high number of layers required for good predictive performance may be due to the larger average length of our input text. As seen in Figure 2, we start the first layer with the node features. As mentioned before, we use BERT embeddings for the node features. Thus, we get our node features, $x_i$ for each word piece or node, $i$ So, $H^0$ is computed with $H_0 = XW_0$, $X$ denotes the BERT embeddings for all tokens. Further on, the GAT layers are stacked with $H^{\ell+1} = GAT(H^\ell, A, \theta^\ell)$. The number of GAT layers is a hyperparameter for our model. At the final layer, from $H^\ell$, we take the set of embeddings corresponding to the two entities being compared: $H^\ell_{ent_a} = \{\vec{h}^\ell_i\}_{i \to ent_a}$ and $H^\ell_{ent_b} = \{\vec{h}^\ell_i\}_{i \to ent_b}$.

Then, we do an average pooling on both $H^\ell_{ent_a}, H^\ell_{ent_b}$ to get $\vec{h_{ent_a}}, \vec{h_{ent_b}}$. As we noted, the College Confidential dataset had many cases where an entity was mentioned multiple times in a text. In such cases, we use the average of the vectors $\vec{h_{ent_i}}$ for each mention of $ent_i$. On the other hand, when the entity was not mentioned, we set the output embedding of that entity as zeroes. Finally, we concatenate these two vectors to get $\vec{z} = \vec{h_{ent_a}} \parallel \vec{h_{ent_b}}$ to get our final layer. We pass $\vec{z}$ to a softmax layer to finish up the model.

## 5 Experimental Results

To compare our result against existing techniques, we choose the following baselines.

**1. Majority:** Predict the majority class in the training set.

**2. Simple NLP features + linear classifier:** Here, we use features that do not depend on pre-trained embeddings. We specifically use n-grams and part-of-speech tagging.

**3. Sentiment + linear classifier** Linear classifier using sentiment score and magnitude as features.

**4. Word embedding based methods:** To make use of word embeddings directly, we use BERT embeddings (Devlin et al. 2019). By taking BERT embeddings of each word, we try different versions and train them on the College Confidential dataset. This method puts an upper limit to the number of tokens at 512 tokens. We try the following approaches: BERTAvg: Average of BERT embeddings for all tokens; BERT-FT: BERT embeddings fine tuned to our task with a linear layer and then averaged; BERT-LSTM: Sequential BERT embeddings fed to an LSTM model.

**5. Sentence embedding + classifier:** Sentence embeddings learn representations for the full sentence based on text spanning in order to keep track of the sequence information of words rather than only averaging the word embeddings. As baselines, we use, we use these two well-known sentence embeddings: InferSent (Conneau et al. 2017) and SimCSE(Gao, Yao, and Chen 2021). As the classifier, we use a linear model and XGBoost (Chen et al. 2015).

**6. Modified ED-GAT (Ma et al. 2020):** ED-GAT creates entity-aware dependency graphs and uses Graph Attention Networks with it for the classification process. This can be thought of as a special case of our MultiSentPref method. To adapt ED-GAT to the multi-sentence setting of the College Confidential dataset we add additional edges between the roots of individual sentences' dependency graphs, so that we have a connected graph.

**MultiSentPref implementation:**

We use the GAT implementation in the Pytorch Geometric (PyG) library for our GAT-based model. We used Huggingface's implementation of BERT to generate our embeddings with the pre-trained weights. The hidden size of each GAT layer is set to 300 with 6 attention heads per layers. For the loss criterion, we used Adam optimizer with learning rate set to $1e-5$. We trained our model with batch size 64 for both training and testing. We got the best performance for 20 layers by tuning for the number of GAT layers (varying between 6 to 30 inclusive at 2 interval). All training was done on Google Colab, with Nvidia K80 GPUs (12 GB).

We show the results of our MultiSentPref model against existing models in Table 3. We see that our model gets the highest macro F1-score and the best F1-score for both preference type. The slightly lower F1(*None*) score indicates that the model trades off performance in predicting the type of preference with predicting whether there is a preference. We notice that while some other models have higher Micro F1-scores, it is mostly because of the higher F1(*None*) scores. It is interesting to see that even with larger pieces of texts, the sentence embedding-based methods do a good job of predicting whether there is a preference or not. Without more advanced methods like using dependency and coreference graph, it is not possible to get better prediction for the actual preferences in a long span of text. Since the College Confidential dataset has texts from two types of discussions (two

| | Models | Macro | Micro | F1($c \succ d$) | F1($c \prec d$) | F1(None) |
|---|---|---|---|---|---|---|
| Baselines | Majority-Class | 0.21 | 0.45 | 0.00 | 0.00 | 0.62 |
| | n-gram+POS | 0.33 | 0.43 | 0.15 | 0.16 | 0.67 |
| | Sentiment score | 0.41 | 0.50 | 0.28 | 0.23 | 0.71 |
| | BERT-Avg | 0.32 | 0.51 | 0.09 | 0.18 | 0.68 |
| | BERT-FT | 0.27 | 0.60 | 0.06 | 0.01 | 0.75 |
| | LSTM-BERT | 0.37 | 0.52 | 0.19 | 0.25 | 0.67 |
| Existing Methods | InferSentLinear | 0.49 | 0.58 | 0.37 | 0.38 | 0.72 |
| | InferSentXGBoost | 0.47 | 0.62 | 0.33 | 0.33 | 0.76 |
| | SimCSELinear | 0.48 | 0.64 | 0.34 | 0.33 | 0.77 |
| | SimCSEXGBoost | 0.51 | **0.67** | 0.39 | 0.35 | **0.79** |
| | ED-GAT | 0.55 | 0.62 | 0.53 | 0.40 | 0.73 |
| Ours | MultiSentPref-15 | 0.53 | 0.58 | 0.49 | 0.42 | 0.69 |
| | MultiSentPref-20 | **0.57** | 0.61 | **0.60** | **0.42** | 0.69 |

Table 3: Comparison of performance of various models on the CPC task on College Confidential Dataset.

| Text Type | macro | F1($c \succ d$) | F1($c \prec d$) | F1(None) |
|---|---|---|---|---|
| Two entities | 0.57 | 0.67 | 0.41 | 0.63 |
| Three entities | 0.59 | 0.57 | 0.45 | 0.75 |

Table 4: Comparison of performance of MultiSentPref model on different type of texts.

entities vs three entities), we check the performance in both types. Table 4 shows that the performance of the model is slightly different in the two types of data. For three-entity texts, it has higher F1(*None*) score, but lower macro F1-score. However, the fact that we have reasonably high predictive performance even for the more complex three-entity texts is encouraging.

## 6 Discussion and Future Work

We see that our models for CPC in multi-sentence, multi-mention-of-entity scenarios outperform the baselines. But the achieved accuracy is not as high as was for CompSent-19, indicating that learning in such more realistic scenarios is a more difficult task. Recently, (Li et al. 2021) showed how knowledge transfer from the task of sentiment analysis helped to improve the predictive performance on the CompSent-19 dataset. Thus, looking into other NLP tasks with a sufficient amount of labeled realistic data related to this problem would be interesting.

## References


Chen, D.; and Manning, C. D. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the EMNLP 2014*, 740–750.

Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K.; et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4).

Clark, K.; and Manning, C. D. 2016a. Deep Reinforcement Learning for Mention-Ranking Coreference Models. In *Proceedings of EMNLP 2016*.

Clark, K.; and Manning, C. D. 2016b. Improving Coreference Resolution by Learning Entity-Level Distributed Representations. arXiv:1606.01323.

Conneau, A.; Kiela, D.; Schwenk, H.; Barrault, L.; and Bordes, A. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of EMNLP 2017*, 670–680. Copenhagen, Denmark: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805.

Dozat, T.; and Manning, C. D. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Gao, T.; Yao, X.; and Chen, D. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of EMNLP 2021*.

Haque, A.; Garg, V.; Guo, H.; and Singh, M. P. 2022. Pixie: Preference in Implicit and Explicit Comparisons. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 106–112.

Lee, K.; He, L.; and Zettlemoyer, L. 2018. Higher-Order Coreference Resolution with Coarse-to-Fine Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 687–692.

Li, Z.; Qin, Y.; Liu, Z.; and Wang, W. 2021. Powering Comparative Classification with Sentiment Analysis via Domain Adaptive Knowledge Transfer. In *Proceedings of the EMNLP 2021*.

Ma, N.; Mazumder, S.; Wang, H.; and Liu, B. 2020. Entity-aware dependency-based deep graph attention network for comparative preference classification. In *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL-2020)*.

Mohsin, F.; Luo, L.; Ma, W.; Kang, I.; Zhao, Z.; Liu, A.; Vaish, R.; and Xia, L. 2021. Making Group Decisions from Natural Language-Based Preferences. In *The 8th International Workshop on Computational Social Choice (COMSOC-2021)*.

Panchenko, A.; Bondarenko, A.; Franzek, M.; Hagen, M.; and Biemann, C. 2019. Categorizing Comparative Sentences. In *Proceedings of the 6th Workshop on Argument Mining*, 136–145. Florence, Italy: Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Proceedings of NeurIPS 2017*.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.