Sentiment Analysis using the Relationship between Users and Products

Natthawut Kertkeidkachorn Kiyoaki Shirai
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{natt, kshirai}@jaist.ac.jp

Abstract

In product reviews, user and product aspects are useful in sentiment analysis. Nevertheless, previous studies mainly focus on modeling user and product aspects without considering the relationship between users and products. The relationship between users and products is typically helpful in estimating the bias of a user toward a product. In this paper, we, therefore, introduce the Graph Neural Network-based model with the pre-trained Language Model (GNNLM), where the relationship between users and products is incorporated. We conducted experiments on three well-known benchmarks for sentiment classification with the user and product information. The experimental results show that the relationship between users and products improves the performance of sentiment analysis. Furthermore, GNNLM achieves state-of-the-art results on yelp-2013 and yelp-2014 datasets.

1 Introduction

Sentiment analysis aims to understand a user's opinion toward a product. It is to infer the sentiment polarity or intensity on a review of a document (Pang et al., 2008; Liu, 2012). Recently, user and product information in a review has been proven to be helpful for sentiment analysis models (Tang et al., 2015). Consequently, many studies investigate how to model user and product aspects and incorporate them into deep neural network models.

Nevertheless, none of them focuses on the relationship between users and products. This relationship between users and products typically provides the bias of a user's sentiment toward a product. For example, users A and B share similar sentiments on many products. If there is a product for which we do not know user A's sentiment, but we know user B's sentiment, we might be able to infer user A's sentiment from user B's sentiment. In addition, if a user has a high expectation toward the product, but the product does not meet the expectation, it would greatly impact the user's sentiment. Meanwhile, the interaction between users and products has proven to be useful in other tasks, such as spam detection (Wang et al., 2012) and citation recommendation (Jeong et al., 2020; Bhowmick et al., 2021). Based on these observations, we assume that the relationship between users and products could provide a clue to help sentiment analysis.

In this paper, we, therefore, propose a new approach using graph neural networks with the pre-trained language model, namely GNNLM. In GNNLM, the relationship between the user and the product is captured by the graph neural network model as distributed representations and then combined with a distributed representation of reviews obtained from a pre-trained language model to predict the sentiment label. We conduct experiments on three benchmarks (IMDB, Yelp-2013, and Yelp-2014) for sentiment classification with the user and product information. The results show that combining the relationship between the user and the product could help improve the performance of the sentiment analysis model.

2 Related Work

Recent studies have shown that user and product information is useful for sentiment analysis. The first study (Tang et al., 2015) argues that user and product information are consistent with a sentiment from a review. They propose UPNN that incorporates the user and product preference matrix into a CNN-based model to modify the meaning of word representation. UPDMN (Dou, 2017) uses a deep memory network to capture the user and product preferences with the LSTM-based model. NSC (Chen et al., 2016) is the model using a hierarchical neural network with the attention mechanism to capture global user and product information. HCSC (Amplayo et al., 2018) investigates the cold start problem for sentiment analysis with the user and product information by introducing shared user

Findings of the Association for Computational Linguistics: ACL 2023, pages 8611–8618
July 9-14, 2023

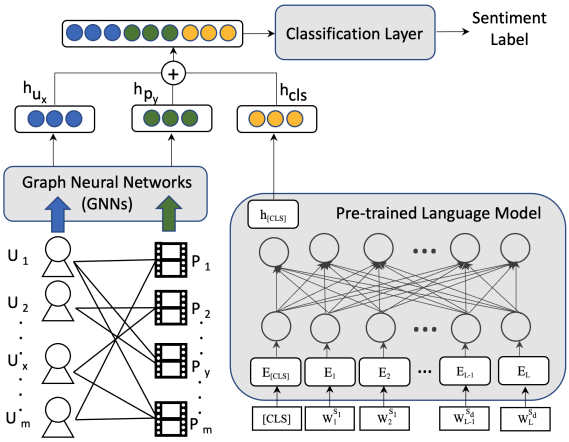

Figure 1: The overall architecture of GNNLM.

and product representations. DUPMN (Long et al., 2018) uses a hierarchical LSTM-based model to encode the document with dual memory networks, one for user information and the other for production information. CMA (Ma et al., 2017) encodes the document using a hierarchical LSTM-based model, in which user and product information are injected hierarchically. BiLSTM + basis-cust (Kim et al., 2019) is a model that combines categorical metadata of users and products into the neural network model. CHIM (Amplayo, 2019) utilizes chunk-wise matrices to represent the user and product aspects and injects them into different locations of the model. IUPC (Lyu et al., 2020) is a model built on stacked attention with BERT to memorize historical reviews of a user and all reviews of a product. MA-BERT (Zhang et al., 2021) is a multi-attribute BERT, where user and product aspects are incorporated into the BERT model.

Based on our survey, none of them investigates the relationship between users and products for sentiment analysis.

3 Our Approach

As shown in Fig. 1, our approach, GNNLM, consists of three components: 1) Graph neural networks, 2) Pre-trained language model, and 3) Classification layer. The task definition and the details of each component are described as follows.

3.1 Task Definition

Sentiment analysis with user and product information is a task to predict the intensity of the polarity of a review using text, user, and product information. The task is defined as follows. Given $U = \{u_1, u_2, u_3, ..., u_n\}$, $P = \{p_1, p_2, p_3, ..., p_m\}$ and R are the set of users, products, and reviews respectively, and a user $u_x \in U$ writes a review $r_{u_x,p_y} \in R$ about the product $p_y \in P$, and r is a review represented by d sentences $\{s_1, s_2, s_3..., s_d\}$ and, the i-th sentence s_i consists of l_i word as $\{w_1, w_2, w_3, ...w_{l_i}\}$, the objective of the task is to model the function $f : (r_{u_x,p_y}, u_x, p_y) \rightarrow \eta;\ \eta \in \mathbb{Z}^+_{[1,K]}$, where η is the polarity scale of the review r_{u_x,p_y} in the Likert scale from 1 to K, and K is the number of polarity classes.

3.2 Graph Neural Networks

Graph Neural Networks (GNNs) are neural models that can capture the dependency between nodes in a graph via message passing (Zhou et al., 2020). Recently, GNNs have been shown effective for various graph-related applications, e.g., Link Prediction (Zhang and Chen, 2018), due to their ability to learn structural information from the graph. In our study, we build the user-product graph and use GNNs to learn structural information representing the relationship between users and products.

In our task, there are two types of nodes: user and product. The user-product graph is defined as the heterogeneous graph $G = (V_U \cup V_P, E)$, where V_U, V_P, and E are the set of user nodes, product nodes, and edges between users and products. All users in U and products in P are used to create user and product nodes. For edges, if user u_x writes a review about the product p_y, there are two edges: (v_{u_x}, v_{p_y}) and (v_{p_y}, v_{u_x}), where $v_{u_x} \in V_U$ and $v_{p_y} \in V_P$. To avoid leaking the structural information between users and products, we only use the training set to build the graph G.

To learn representations of users and products, we use GraphSAGE (Hamilton et al., 2017) as the graph neural network operator to aggregate the structure information of the graph G. One advantage of GraphSAGE is that it can leverage the topological structure of neighbor nodes to learn and generalize embeddings of unseen nodes. Formally, the representation of nodes in the graph G is computed as follows:

$$h^i_{\mathcal{N}_v} = aggregate(h^{i-1}_u, \forall u \in \mathcal{N}_v) \quad (1)$$

$$h^i_v = \sigma(W^i \cdot [h^{i-1}_v; h^i_{\mathcal{N}_v}]) \quad (2)$$

where $aggregate(\cdot)$ is the function to aggregate information from neighbor nodes, $\sigma(\cdot)$ is the activation function, $\mathcal{N}_v$ is a set of all neighbor nodes of the node v, W^i is a set of weight matrices used to propagate information between different layers,

and h_v^i is the representation of the node v at the i-th layer. By computing representations of all nodes, we could encode the relationship between the user and the product as the vector representation.

3.3 Pre-trained Language Model

Pre-trained language models, such as BERT (Devlin et al., 2019) and RoBERTa(Liu et al., 2019), can achieve remarkable performance for many NLP tasks by the fine-tuning method. In our study, we use the pre-trained language model to learn the representation of a review. Using a word piece tokenizer (Wu et al., 2016), the review r_{u_x,p_y} can be represented as a sequence of tokens $c_{r_{u_x,p_y}} = \{[CLS], w_1^{s_1}, w_2^{s_1}, ..., w_1^{s_2}, ..., w_{l_d}^{s_d}\}$, where $[CLS]$ is a special token representing the whole sequence. To obtain the representation of review r_{u_x,p_y}, we feed the sequence $c_{r_{u_x,p_y}}$ into the pre-trained language model as follows.

$$h_{cls} = f_{LM}(c_{r_{u_x,p_y}}; \theta_{LM}) \tag{3}$$

where $f_{LM}(\cdot)$ is the pre-trained language model, and θ_{LM} is its trainable parameters initialized from the pre-trained language model checkpoint.

3.4 Classification Layer

The classification layer is the final layer that combines the representation of the review r_{u_x,p_y} with the representations of the user u_x and the product p_y to predict the intensity of the polarity. In the classification layer, the representations of r_{u_x,p_y}, u_x, and p_y are concatenated and then passed into a feed-forward neural network with a rectified linear unit $(ReLU)$ function to project them into the target space of polarity classes. The classification layer can be defined as:

$$\hat{p} = ReLU(W_K \cdot [h_{cls}; h_{u_x}; h_{p_y}] + b_K) \tag{4}$$

where h_{cls} is the representation of review r_{u_x,p_y} from the pre-trained language model, h_{u_x} and h_{p_y} are the representations of user u_x and product p_y from GNNs, W_K and b_K are the parameters of the neural network. Then, the softmax function in Eq. 5 is used to normalize the polarity distribution.

$$\hat{y} = \frac{exp(\hat{p})}{\sum_{i=1}^{K} exp(\hat{p}_i)} \tag{5}$$

where K is the number of polarity classes.

To learn and optimize our model, we use a cross-entropy loss function defined as follows:

$$L = -\sum_{r \in R} \sum_{i=1}^{K} y_{r,i} \cdot log(\hat{y}_{r,i}) \tag{6}$$

where $y_{r,i}$ represents agreement with the ground-truth. Its value is 1 if the gold polarity class of the review r is i; otherwise 0.

4 Experiment

4.1 Experimental Setup

Setting. The experimental setting follows the same setting in the study (Tang et al., 2015). In the setting, there are three benchmarks: IMDB, Yelp-2013, and Yelp-2014. The evaluation metrics are accuracy (Acc), and root mean squared error (RMSE).

Implementation. In GNNLM, we implement GNNs by using SAGEConv (Hamilton et al., 2017) and the pre-trained language model by using the RoBERTa (Liu et al., 2019) from Huggingface (Wolf et al., 2020). Note that in our preliminary experiment using the pre-trained language models, we were unable to reproduce the results for BERT as reported in (Lyu et al., 2020; Zhang et al., 2021) on the IMDB dataset. However, we could achieve comparable results as presented in (Lyu et al., 2020) by utilizing RoBERTa. To ensure fairness in the evaluation, we therefore selected RoBERTa as the pre-trained language model. The dimension of each node in GNNs and the dimension of hidden representations of RoBERTa are 768. The maximum sequence length of RoBERTa is 512. The AdamW optimizer (Loshchilov and Hutter, 2017) is used with the learning rate set at 2e-5. The batch size is set to 32. In the fine-tuning process, the model is trained up to 10 epochs on the training set. We select the best hyper-parameters from the dev set for evaluation in the test set. The source code and the setting for the experiments are available on the GitHub repository.[1]

While we can simply fine-tune the pre-trained language model, the user and product representations from GNNs are randomly initialized and needs to be trained from scratch. To better learn the user and product representations before combing them, we train GNNLM with only GNNs for 100 epochs on the training set and save it as the GNNs checkpoint. In the fine-tuning process, the RoBERTa checkpoint and GNNs checkpoint are loaded to initialize the models.

[1] https://github.com/knatthawut/gnnlm

Methods	IMDB		Yelp-2013		Yelp-2014	
	Acc	RMSE	Acc	RMSE	Acc	RMSE
Majority (Tang et al., 2015)	19.6	2.495	39.2	1.097	41.1	1.060
BERT (IUPC) (Lyu et al., 2020)	47.9	1.243	67.2	0.647	67.5	0.621
BERT (MA-BERT) (Zhang et al., 2021)	51.8	1.191	67.7	0.627	67.2	0.630
UPNN (Tang et al., 2015)	43.5	1.602	59.6	0.803	60.8	0.764
UPDMN (Dou, 2017)	46.5	1.351	61.3	0.720	63.9	0.662
NSC (Chen et al., 2016)	53.3	1.281	65	0.692	66.7	0.654
HCSC (Amplayo et al., 2018)	54.2	1.213	65.7	0.660	-	-
DUPMN (Long et al., 2018)	53.9	1.279	66.2	0.667	67.6	0.639
CMA (Ma et al., 2017)	54.0	1.191	66.4	0.677	67.6	0.637
BiLSTM+basis-cust (Kim et al., 2019)	-	-	67.1	0.662	-	-
CHIM (Amplayo, 2019)	56.4	1.161	67.8	0.646	69.2	0.629
IUPC (Lyu et al., 2020)	53.8	1.151	70.5	0.589	71.2	0.592
MA-BERT (Zhang et al., 2021)	**57.3**	**1.042**	70.3	0.588	71.4	0.573
ISAR (Wen et al., 2023)	56.6	1.186	69.1	0.619	69.3	0.621
GNNLM-GNNs	32.6	2.095	46.7	1.094	46.2	1.108
GNNLM-LM	48.3	1.191	67.2	0.618	67.3	0.616
GNNLM	54.4	1.102	**72.2**	**0.573**	**72.1**	**0.568**

Table 1: Experimental Results on IMDB, Yelp-2013, and Yelp-2014

For the ablation study, we also evaluate GNNs and RoBERTa separately. GNNLM-GNNs denotes our model with only GNNs, while GNNLM–LM refers to our model with only RoBERTa.

Baseline. We compare our GNNLM with all systems from the leaderboard[2] for this task. On the leaderboard, there are 10 systems: UPNN (Tang et al., 2015), UPDMN (Dou, 2017), NSC (Chen et al., 2016), HCSC (Amplayo et al., 2018), DUPMN (Long et al., 2018), CMA (Ma et al., 2017), BiLSTM+basis-cust (Kim et al., 2019), CHIM (Amplayo, 2019), IUPC (Lyu et al., 2020) and MA-BERT (Zhang et al., 2021). In addition, we conduct a comparison between our approach and ISAR (Wen et al., 2023), a recently published baseline that employs graph ranking to model the interaction between users and products.

Moreover, we use three additional baselines: Majority (Tang et al., 2015), BERT (UPIC) (Lyu et al., 2020), and BERT (MA-BERT) (Zhang et al., 2021). Majority always chooses the polarity class based on the majority labels in the training set. Both BERT (UPIC) and BERT (MA-BERT) are BERT models.

4.2 Result and Discussion

The experimental results are listed in Table 1. Considering our variations of GNNLM models, we found that GNNLM outperforms GNNLM-GNNs and GNNLM-LM. It infers that the representation learned from the relationship between users and products could help improve the performance of sentiment analysis.

GNNLM-GNNs mostly achieves better results than Majority. Majority could be considered as the heuristic approach using the majority polarity between users and products. From the results, GNNLM-GNNs could encode structural information, which is more useful than the majority polarity between users and products. Nonetheless, GNNLM-GNNs could suffer from the sparsity problem. The density of the user-product graph on IMDB, Yelp-2013, and Yelp-2014 is 0.06, 0.05, and 0.02. The graph in Yelp-2014 is sparser than the others. This sparsity problem could be the reason for no improvement in RMSE of GNNLM-GNNs compared with Majority. To further study the impact of the sparsity problem, we analyze the results based on the degree of a node in the graph. We found that nodes with lower degrees tend to provide lower performance. Therefore, the sparsity impacts the performance of GNNLM-GNNs.

Comparing our GNNLM with the systems on the leaderboard, we found that GNNLM could achieve the best performance on the Yelp-2013 and Yelp-2014 datasets. For the IMDB dataset, GNNLM could outperform most systems, except

[2] http://nlpprogress.com/english/sentiment_analysis.html

for MA-BERT in both metrics and CHIM, ISAR in the Acc metric. GNNLM could not surpass MA-BERT due to the performance of the base model. GNNLM-LM, BERT (IUPC), and BERT (MA-BERT) are pre-trained language models without the user and product information. On the Yelp-2013 and Yelp-2014 datasets, the performances of these approaches are comparable; however, on the IMDB dataset, BERT (MA-BERT) significantly outperforms GNNLM-LM and BERT (IUPC). Therefore, the large difference in the base model's performance could be the main reason for the gap between GNNLM and MA-BERT on the IMDB dataset.

5 Conclusion

This paper introduces GNNLM, GNNs with the pre-trained language model for sentiment analysis with user and product information. Unlike previous studies, we incorporate the relationship between users and products into the model using GNNs. Experimental results show that the representations learned from the relationship between users and products contribute to sentiment analysis models. In the future, we will attempt to model user and product aspects from reviews into the graph.

Limitations

Our approach relies on the pre-trained language model performance. Although using a graph neural network with the user-product graph helps improve the performance in sentiment analysis, the pre-trained language model still plays an important role in the task. If the pre-trained language model cannot obtain good results, it will affect the performance as discussed on the IMDB dataset.

Furthermore, the graph density could affect the performance of GNNLM-GNNs, as discussed in the experimental results. Since GNNLM is built on top of GNNLM-GNNs, GNNLM is also affected by the sparsity problem. As already reported, the density of the user-product graph on the IMDB, Yelp-2013, and Yelp-2014 datasets are 0.06, 0.05, and 0.02, respectively. The greater the value is, the denser the graph is. Comparing GNNLM with GNNLM-LM, we found that the improvements we could obtain on the IMDB, Yelp-2013, and Yelp-2014 datasets are 6.1, 5.0, and 4.8, respectively. The trend of improvement conforms with the density of the graph. Therefore, if the user-product graph is very sparse, it would greatly affect the performance of GNNLM.

References

Reinald Kim Amplayo. 2019. Rethinking attribute representation and injection for sentiment classification. 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 5602–5613, Hong Kong, China. Association for Computational Linguistics.

Reinald Kim Amplayo, Jihyeok Kim, Sua Sung, and Seung-won Hwang. 2018. Cold-start aware user and product attention for sentiment classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2535–2544, Melbourne, Australia. Association for Computational Linguistics.

Anubrata Bhowmick, Ashish Singhal, and Shenghui Wang. 2021. Augmenting context-aware citation recommendations with citation and co-authorship history. In *18th International Conference on Scientometrics and Informetrics, ISSI 2021*, pages 115–120. International Society for Scientometrics and Informetrics.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659, Austin, Texas. Association for Computational Linguistics.

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

Zi-Yi Dou. 2017. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 521–526, Copenhagen, Denmark. Association for Computational Linguistics.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Chanwoo Jeong, Sion Jang, Eunjeong Park, and Sungchul Choi. 2020. A context-aware citation recommendation model with bert and graph convolutional networks. *Scientometrics*, 124:1907–1922.

Jihyeok Kim, Reinald Kim Amplayo, Kyungjae Lee, Sua Sung, Minji Seo, and Seung-won Hwang. 2019.

Categorical metadata representation for customized text classification. *Transactions of the Association for Computational Linguistics*, 7:201–215.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yunfei Long, Mingyu Ma, Qin Lu, Rong Xiang, and Chu-Ren Huang. 2018. Dual memory network model for biased product review classification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 140–148, Brussels, Belgium. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Chenyang Lyu, Jennifer Foster, and Yvette Graham. 2020. Improving document-level sentiment analysis with user and product context. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6724–6729, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, and Xu Sun. 2017. Cascading multiway attentions for document-level sentiment classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 634–643, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1014–1023, Beijing, China. Association for Computational Linguistics.

Guan Wang, Sihong Xie, Bing Liu, and Philip S Yu. 2012. Identify online store review spammers via social review graph. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(4):1–21.

Jiahui Wen, Anwen Huang, Mingyang Zhong, Jingwei Ma, and Youcai Wei. 2023. Hybrid sentiment analysis with textual and interactive information. *Expert Systems with Applications*, 213:118960.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

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

Muhan Zhang and Yixin Chen. 2018. Link prediction based on graph neural networks. *Advances in neural information processing systems*, 31.

You Zhang, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2021. MA-BERT: Learning representation by incorporating multi-attribute knowledge in transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2338–2343, Online. Association for Computational Linguistics.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. 2020. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations Section

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Experimental Setup

☑ B1. Did you cite the creators of artifacts you used?
Experimental Setup

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Experimental Setup

C ☑ Did you run computational experiments?

Experimental Setup

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Not applicable. Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Experimental Setup

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Experimental Setup

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.