Robust Learning for Multi-party Addressee Recognition with Discrete Addressee Codebook

Pengcheng Zhu, Wei Zhou, Kuncai Zhang, Yuankai Ma, Haiqing Chen
Alibaba Group
{tangju.zpc, fayi.zw, kuncai.zkc, yuankai.myk, haiqing.chenhq}@alibaba-inc.com

Abstract

Addressee recognition aims to identify addressees in multi-party conversations. While state-of-the-art addressee recognition models have achieved promising performance, they still suffer from the issue of robustness when applied in real-world scenes. When exposed to a noisy environment, these models regard the noise as input and identify the addressee in a pre-given addressee closed set, while the addressees of the noise do not belong to this closed set, thus leading to the wrong identification of addressee. To this end, we propose a Robust Addressee Recognition Model (RARM), which discretizes the addressees into a codebook, making it able to represent addressees in the noise and robust in a noisy environment. Experimental results show that the introduction of the addressee codebook helps to represent the addressees in the noise and highly improves the robustness of addressee recognition even if the input is noise.

1 Introduction

Different from two-party conversation, multi-party conversation has more than two interlocutors (Traum, 2003; Uthus and Aha, 2013; Meng et al., 2018; Gu et al., 2021). Beyond response generation or selection (Hu et al., 2019; Liu et al., 2019; Gu et al., 2020; Wang et al., 2020b), there is also a need for recognizing the addressee of the multi-party conversation (Ouchi and Tsuboi, 2016; Zhang et al., 2018; Le et al., 2019).

Addressee recognition aims to identify the interlocutors indicate to whom they are speaking. Ouchi and Tsuboi (2016) formalize the task as given a context to predict an addressee, the system is required to select an addressee appearing in the previous context. Meng et al. (2018) realize the importance of speaker modeling and propose speaker classification as a surrogate task for general speaker modeling. Zhang et al. (2018) use a novel dialogue encoder to update speaker embeddings in a role-sensitive way. Le et al. (2019) not only focuses on predicting the addressee of the last utterance but also aims to predict all the missing addressees. Gu et al. (2021) propose a unified multi-party pretrain model and design five self-supervised tasks based on the interactions among utterances and interlocutors.

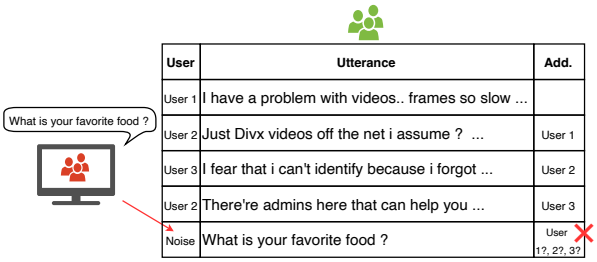


Figure 1: Example of multi-party conversation in a noisy environment.

These works suppose that the multi-party conversation happens in a quiet environment, which can lead to serious system failure when exposed to a noisy environment. Many other works recently focus on robust learning in practice (Wang et al., 2020a; Xue et al., 2020; Liu et al., 2021; Wang et al., 2022). However, these robust learning works mainly focus on two-party conversations and introduce noises by replacing, inserting, swapping, and deleting characters at the word level or words at the sentence level. The main difference between two-party conversation and multi-party conversation is that two-party conversation mainly focuses on perturbations at the semantic level, while beyond semantic perturbation, the multi-party conversation should consider the perturbations that are not intended for the current conversation, even if the noise is semantically complete. As shown in Figure 1, the noise is semantically complete but doesn't belong to the current conversation.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 571–578

Since the number of addressees in a noisy environment is unknowable, giving a fixed length of the addressee matrix is not feasible. On account of the above issues, we propose the Robust Addressee Recognition Model (RARM), which discretizes the addressees into a codebook and represents addressees by addressee codes. We evaluate our method on two types of addressee noise: in-domain addressee noise (ID-AN) and out-domain addressee noise (OD-AN). The ID-AN is the noise that has the same domain as the current multi-party conversation, and OD-AN is the noise that doesn't have.

The main contributions are as follows: (1) We formalize the task of Robust Addressee Recognition (RAR) task in multi-party conversation and propose the Robust Addressee Recognition Model (RARM), which discretizes the addressees into a codebook, making addressee recognition robust in a noisy environment. (2) We conduct experiments on two types of noise: in-domain and out-domain noise, experimental results show that the addressee codebook helps to represent the addressees in noise effectively and highly improves the robustness of addressee recognition even if the input is in-domain or out-domain noise.

2 Methods

2.1 Task Definition

We follow Ouchi and Tsuboi (2016) to define the addressee recognition. Given a multi-party conversation S, the task is to select an addressee for the last utterance q in the candidate set A.

$$GIVEN \quad : \quad S = (q, C) \tag{1}$$
$$PREDICT \quad : \quad \hat{a} \in A \tag{2}$$

where C is context. When considering noise N, the formulation of robust addressee recognition is updated as:

$$GIVEN \quad : \quad S = (q, C) \tag{3}$$
$$PREDICT \quad : \quad \hat{a} \in \{A, N\} \tag{4}$$

2.2 Robust Addressee Recognition Model

One straightforward way to represent the noise is to add an extra vector in the addressee matrix, while it is too rough to represent all addressees in the noise into the same vector. In this section, we propose to utilize VQ-VAE (van den Oord et al., 2017) to discretize addressees into a codebook.

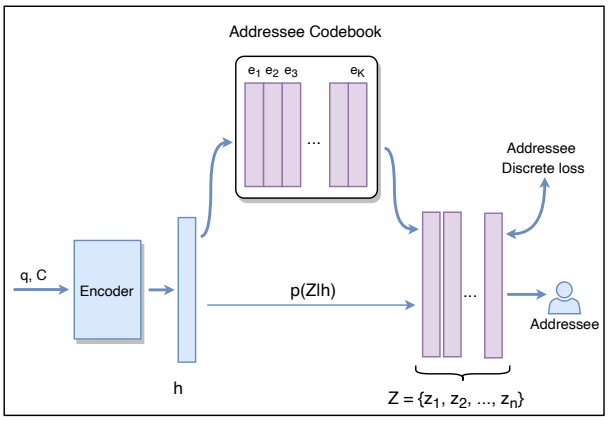


Figure 2: The architecture of the proposed method.

There are three parts in the RARM: an encoder for query and context representation, a discrete addressee codebook for addressee representation, and a classifier for addressee classification. The model architecture is illustrated in Figure 2.

2.3 Encoder

We use Transformer (Vaswani et al., 2017) with 12 layers as Encoder, and the input is the concatenation of query q and context C with special token '[SEP]'. The representation of the input sequence is defined as:

$$h = Transformer(q, C) \tag{5}$$

where h is the hidden states at the position of special token '[CLS]'.

2.4 Discrete Addressee Codebook

Addressee codebook is an embedding table $e \in R^{K*d}$ where K is discrete latent variables size. We follow van den Oord et al. (2017) to discretize the addressee into the codebook as follows:

$$p(z_1|h) = \begin{cases} 1 \;\; if \; k = \arg\min_j ||h - e_j||^2 \\ 0 \;\; otherwise \end{cases} \tag{6}$$

thus h is mapped onto the embedding e_k as:

$$z_1 = e_k, \quad where \; k = \arg\min_j ||h - e_j||^2 \tag{7}$$

In order to augment the representation of addresses, we discretize an addressee into a code set Z instead of one code. The difference between h and z_1 is fed back to the discrete process and repeats the steps above n times as follows:

$$h_1 = h - z_1 \tag{8}$$
$$z_2 = Discrete(h_1) \tag{9}$$

Types	Hu et al. (2019)	Ouchi and Tsuboi (2016)		
		length-5	length-10	length-15
ID-AN	93517 / 1500 / 1500	138336 / 8571 / 9800	148567 / 9292 / 10691	146943 / 9244 / 10615
OD-AN	93517 / 1500 / 1500	138336 / 8571 / 9800	148567 / 9292 / 10691	146943 / 9244 / 10615
Overall	311725 / 5000 / 5000	461120 / 28570 / 32668	495226 / 30974 / 35638	489812 / 30815 / 35385

Table 1: The statistics of the constructed ID-AN, OD-AN, and Overall data in the dataset.

Models	Types	Hu et al. (2019)	Ouchi and Tsuboi (2016)		
			length-5	length-10	length-15
BERT	ID-AN	59.3	51.3	46.6	46.2
	OD-AN	81.7	74.5	70.1	68.9
	Overall	80.4	71.7	67.3	66.8
MPC-BERT	ID-AN	64.6	56.1	53.3	51.8
	OD-AN	84.4	77.4	74.8	73.5
	Overall	83.8	75.1	72.6	70.4
RARM w/o codebook	ID-AN	61.3	53.8	50.4	48.6
	OD-AN	82.6	75.8	72.7	70.3
	Overall	81.5	72.8	70.1	68.4
RARM w/o AD loss	ID-AN	65.9	57.7	54.5	**52.9**
	OD-AN	85.1	**79.4**	75.8	74.1
	Overall	84.5	76.3	72.7	71.1
RARM	ID-AN	**67.7**	**58.3**	**55.2**	52.6
	OD-AN	**86.4**	79.2	**77.6**	**74.7**
	Overall	**85.1**	**76.9**	**73.8**	**71.5**

Table 2: Automatic evaluation results on the dataset. ID-AN/OD-AN means performances only on ID-AN/OD-AN data. Overall means overall performance on all clean, ID-AN, and OD-AN data.

where $Discrete$ means the discrete process in equation (6) and (7). Thus the final representation of an addressee is computed as:

$$Z = \{z_1, z_2, ..., z_n\} \tag{10}$$

we set $n = 3$ in all experiments. Thus a resulting addressee is selected as follows:

$$P_a = Softmax\left(W([z_1 : z_2 : z_3]) + b\right) \tag{11}$$

$$\hat{a} = \underset{a \in \{A,N\}}{Argmax}\left(P_a\right) \tag{12}$$

2.5 Training

The RARM is trained with two losses as follows:

Classification loss with VQ-VAE aims to train the codebook and recognize addressees with the code. We follow (van den Oord et al., 2017) to train our model with loss defined as follows:

$$\begin{aligned} loss_{vq} = -logp(y|Z) + ||Z - sg[h]||_2^2 \\ +\beta||h - sg[Z]||_2^2 \end{aligned} \tag{13}$$

where sg stands for the stopgradient operator that has zero partial derivatives. The first term is classification loss. The middle term is the codebook loss that optimizes the codebook embedding. Encoder is optimized by the first and the last term, and we set $\beta = 0.25$ in all experiments.

Addressee Discrete loss is utilized to discretize addressee into the codebook. Zhao et al. (2017) has proved the effect of bag-of-words (BOW) loss on discrete latent variables, we define the addressee discrete loss as follows:

$$loss_{discrete} = -\sum_{i=0}^{y_q} logp(y_i|Z) \tag{14}$$

where y_q is the query words set.

3 Experiment

3.1 Experimental Setups and Dataset

We evaluated our proposed methods on Ubuntu IRC benchmarks (Le et al., 2019) and (Ouchi and Tsuboi, 2016). We define two types of addressee noise: in-domain addressee noise (ID-AN) and out-domain addressee noise (OD-AN). The ID-AN is the noise that has the same domain as the current multi-party conversation, and OD-AN is the noise that doesn't have.

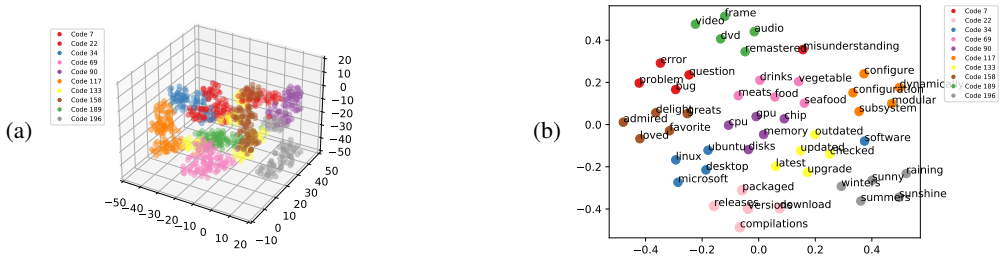


Figure 3: (a): Visualization of the word embeddings that are close to the codes with 3D T-SNE. (b): Visualization of the sampled embeddings that are close to the codes with 2D T-SNE.

For the construction of ID-AN, we replace the query by sampling a query in another conversation in the same Ubuntu benchmark. As for OD-AN, we replace the query by sampling a query in the DailyDialog dataset (Li et al., 2017), which is a high-quality chit-chat dialog dataset. The statistics of the constructed ID-AN, OD-AN, and Overall data in the dataset are shown in Table 1. We follow the splitting strategy of Gu et al. (2021) and set the clean / ID-AN / OD-AN at the ratio of 40%/30%/30% in train/dev/test set.

We set codebook size K to 200 and the dimension of embedding vector d to 768. The weight of the addressee discrete loss is 0.02. The checkpoint with the lowest loss on the validation set is selected for testing.

We compare RARM with baselines: (1) **BERT** is a pretrain bidirectional transformers classification model with self-attention (Devlin et al., 2019). (2) **MPC-BERT** is a pretrained language model for multi-party conversation understanding, which achieves SOTA performance in multi-party addressee recognition (Gu et al., 2021).

3.2 Automatic Evaluation

We follow Ouchi and Tsuboi (2016) to evaluate the task with accuracy, three types of results are listed in Table 2, ID-AN/OD-AN means performances only on ID-AN/OD-AN data, and Overall means overall performance on all data with ID-AN/OD-AN.

As shown in the table, our proposed RARM achieves the best performance compared with baselines. Though MPC-BERT achieves SOTA in the addressee recognition task (Gu et al., 2021), it fails to keep the robustness in in-domain and out-domain noisy data. We observe that the performance of ID-AN is much worse than OD-AN, that's because in-domain noise is closer to the conversation compared to out-domain noise, it is hard to distinguish noise in the same domain.

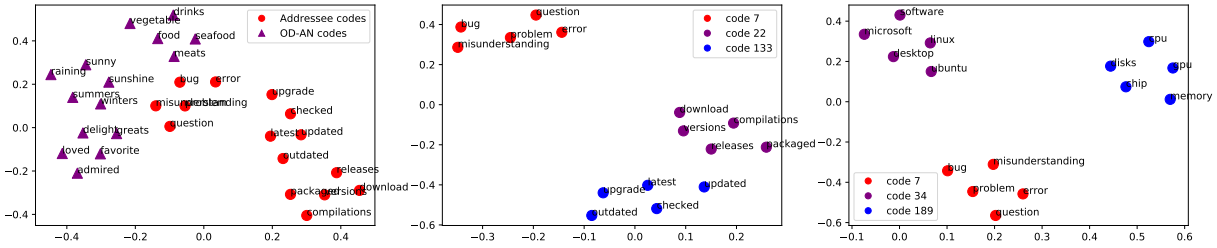
Ablation study results are listed in Table 2. We find that the performance decreases significantly without codebook, demonstrating the importance of discrete of the addressee codebook. The performance drops without addressee discrete loss, mainly because BOW loss helps to represent discrete latent variables.

3.3 Analysis on Addressee Codebook

We sample 10 codes for visualization in Figure 3. We calculate the word embeddings that are close to the sampled codes by cosine similarity and visualize them in 3(a). We find that different codes represent different semantic clusters. To further study the meaning of each code, we sample and visualize five embeddings for each code in 3(b). The figure shows that code 34 (dots in blue) is close to 'ubuntu', 'linux', and 'microsoft', which represent the words related to the operating system. Similarly, code 90 (dots in purple) is close to 'CPU', 'disks', and 'memory', which are related to disk storage capacity.

3.4 Analysis on Addressee Representation

We randomly sample addressees in clean/ID-AN/OD-AN data and visualize corresponding codes in Figure 4. We visualize the word embeddings that are close to the addressee codes in clean and OD-AN data in Figure 4(a). Since we conduct experiments on Ubuntu datasets, the addressee codes in clean data are discretized to Ubuntu-related words, e.g., 'bug', 'upgrade', and 'package', while the correlation of addressee codes in OD-AN with Ubuntu is small, e.g., 'sunny' and 'seafood'. We find that it's easy to distinguish addressees in multi-party conversations from out-domain noise since they don't share the same



(a) (b) (c)

Figure 4: (a)Visualization of sampled OD-AN codes and addressee codes with 2D T-SNE. (b)Visualization of sampled ID-AN codes with 2D T-SNE. (c)Visualization of sampled addressee codes with 2D T-SNE.

codes.

We visualize the word embeddings that are close to the addressee codes in clean and ID-AN data in Figure 4(b) and 4(c). The figures show that clean and ID-AN addressees share the same codes, e.g., code 7, because the ID-AN is also sampled from the Ubuntu IRC datasets, that is, in the same domain. Though it is difficult to distinguish the addressee in clean and ID-AN data at the code semantic level, we observe that the cosine similarity between codes in clean data is smaller than codes in ID-AN data. Code 22 and code 133 in 4(b) mainly represent 'version' and 'upgrade', we can easily infer that the addressee mainly discusses the problem of version upgrade. While code 34 and code 189 represent operating system and disk storage capacity respectively in 4(c), the correlation between code 34 and code 189 is relatively small.

4 Conclusion

In this paper, to improve the robustness of multi-party addressee recognition, we formalize the Robust Addressee Recognition (RAR) task and propose the Robust Addressee Recognition Model (RARM), which discretizes the addressees into a codebook, making it able to represent addressees in noise. We evaluate our method in two types of addressee noise: ID-AN and OD-AN. Experimental results demonstrate that the addressee codebook helps to represent the addressees in noise effectively and highly improves the robustness of addressee recognition even if the input is in-domain or out-domain noise.

5 Limitations

The main limitation is that the in-domain noise is hard to recognize in noisy multi-party conversations. Though our proposed RARM achieves the best performance compared to all baselines, we find that if the content of the noise is close to the multi-party conversation's content, the average accuracy of all methods is not high, how to improve the performance on these hard samples is worthy of further study.

References

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*, pages 4171–4186.

Jia-Chen Gu, Tianda Li, Quan Liu, Zhen-Hua Ling, Zhiming Su, Si Wei, and Xiaodan Zhu. 2020. Speaker-aware bert for multi-turn response selection in retrieval-based chatbots. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2041–2044.

Jia-Chen Gu, Chongyang Tao, Zhenhua Ling, Can Xu, Xiubo Geng, and Daxin Jiang. 2021. Mpc-bert: A pre-trained language model for multi-party conversation understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 3682–3692.

Wenpeng Hu, Zhangming Chan, Bing Liu, Dongyan Zhao, Jinwen Ma, and Rui Yan. 2019. Gsn: A graph-structured network for multi-party dialogues. In *In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, page 5010–5016.

Ran Le, Wenpeng Hu, Mingyue Shang, Zhenjun You, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019. Who is speaking to whom? learning to identify utterance addressee in multi-party conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing, pages 1909–1919.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, pages 986–995.

Cao Liu, Kang Liu, Shizhu He, Zaiqing Nie, and Jun Zhao. 2019. Incorporating interlocutor-aware context into response generation on multi-party chatbots. In *Proceedings of the 23rd Conference on Computational Natural Language Learning*, pages 718–727.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. Robustness testing of language understanding in task-oriented dialog. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 2467–2480.

Zhao Meng, Lili Mou, and Zhi Jin. 2018. Towards neural speaker modeling in multi-party conversation: The task, dataset, and models. In *Proceedings of the Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence*, volume 32.

Hiroki Ouchi and Yuta Tsuboi. 2016. Addressee and response selection for multi-party conversation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2133–2143.

David Traum. 2003. Issues in multiparty dialogues. In *Workshop on Agent Communication Languages*, pages 201–211. Springer.

David C Uthus and David W Aha. 2013. Multiparticipant chat analysis: A survey. *Artificial Intelligence*, pages 106–121.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, page 6306–6315.

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.

Jiayi Wang, Rongzhou Bao, Zhuosheng Zhang, and Hai Zhao. 2022. Distinguishing non-natural from natural adversarial samples for more robust pretrained language model. In *Findings of the Association for Computational Linguistics*, pages 905–915.

Longshaokan Wang, Maryam Fazel-Zarandi, Aditya Tiwari, Spyros Matsoukas, and Lazaros Polymenakos. 2020a. Data augmentation for training dialog models robust to speech recognition errors. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 63–70.

Weishi Wang, Steven CH Hoi, and Shafiq Joty. 2020b. Response selection for multi-party conversations with dynamic topic tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 6581–6591.

Haiyang Xue, Yang Feng, Shuhao Gu, and Wei Chen. 2020. Robust neural machine translation with asr errors. In *Proceedings of the First Workshop on Automatic Simultaneous Translation*, pages 15–23.

Rui Zhang, Honglak Lee, Lazaros Polymenakos, and Dragomir Radev. 2018. Addressee and response selection in multi-party conversations with speaker interaction rnns. In *Proceedings of the Association for the Advancement of Artificial Intellige Conference on Artificial Intelligence*, page 5690–5697.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 654–664.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section 5

☑ A2. Did you discuss any potential risks of your work?
section 5

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

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

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ 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)?
No response.

☐ 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?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ 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.
No response.

C ☑ Did you run computational experiments?

section 3.2 3.3 3.4

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 3.1

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?
section 3.1

☑ 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?
section 3.2

☑ 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.)?
section 3.3

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.