Zero-Shot and Few-Shot Stance Detection on Varied Topics via Conditional Generation

Haoyang Wen and **Alexander G. Hauptmann**
Language Technologies Institute, Carnegie Mellon University
{hwen3, alex}@cs.cmu.edu

Abstract

Zero-shot and few-shot stance detection identify the polarity of text with regard to a certain target when we have only limited or no training resources for the target. Previous work generally formulates the problem into a classification setting, ignoring the potential use of label text. In this paper, we instead utilize a conditional generation framework and formulate the problem as denoising from partially-filled templates, which can better utilize the semantics among input, label, and target texts. We further propose to jointly train an auxiliary task, target prediction, and to incorporate manually constructed incorrect samples with unlikelihood training to improve the representations for both target and label texts. We also verify the effectiveness of target-related Wikipedia knowledge with the generation framework. Experiments show that our proposed method significantly outperforms several strong baselines on VAST, and achieves new state-of-the-art performance.[1]

1 Introduction

Stance detection is an important task that identifies the polarity of text with regard to certain target (Somasundaran and Wiebe, 2010; Augenstein et al., 2016; Mohammad et al., 2016; Sobhani et al., 2017; Allaway and McKeown, 2020), as shown in Table 1. It is crucial for understanding opinionated information expressed in natural language, and it can facilitate downstream social science analyses and applications (Zhang et al., 2017; Hanselowski et al., 2018; Jang and Allan, 2018).

Previous work on stance detection mostly focuses on in-domain or leave-out targets with only a few target choices (Mohtarami et al., 2018; Xu et al., 2018; Graells-Garrido et al., 2020; Zhang et al., 2020; Liang et al., 2021; Allaway et al., 2021; Jiang et al., 2022). Although achieving promising performance, those models are limited to generalize to a wide variety of targets. Zero-shot and few-shot stance detection on varied topics (VAST; Allaway and McKeown, 2020), instead, provides a diverse set of targets for training and testing. Efforts on this direction includes involving graph modeling (Lin et al., 2021), common sense (Liu et al., 2021) or Wikipedia knowledge (He et al., 2022), and contrastive learning (Liang et al., 2022a,b). These methods generally formulate the problem into a classification setting, which directly trains the label representation from scratch, and does not fully utilize the semantics from those label and target texts.

Input Text: Airports and the roads on east nor west coast can not handle the present volume adequately as is. I did ride the vast trains in Europe, Japan and China and found them very comfortable and providing much better connections and more efficient.
Target: high-speed rail **Stance Label:** Supportive (Pro)

Table 1: A stance detection example from VAST.

However, connections among text semantics from input text, target, and label can be beneficial for stance detection. In this paper, we propose a new model by formulating the problem as a denoising task from text templates via conditional generation. Compared to direct classification, we can further exploit the label and topic semantics via learning to decode a series of natural language text containing the predicted label. The denoising scheme can also take advantage of the pretrained language model with similar pretraining task formulation (Lewis et al., 2020). To improve the target representation, we propose to jointly train target prediction with stance detection, which gives the input text and desired stance label to output possible targets. We use unlikelihood training (Welleck et al., 2020) that suppress the likelihood of manually constructed incorrect samples to enhance label

[1]The resource for reproducing this paper is available at https://github.com/wenhycs/ACL2023-Zero-Shot-and-Few-Shot-Stance-Detection-on-Varied-Topics-via-Conditional-Generation.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 1491–1499

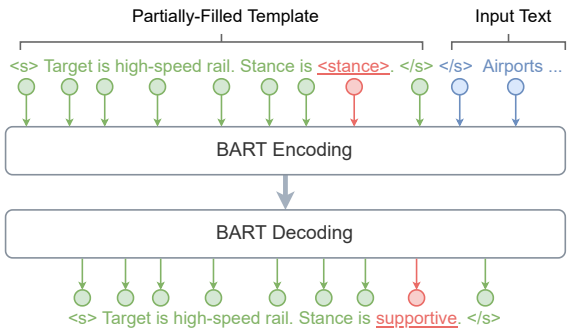


Figure 1: Overall framework of BART-based generation framework for stance detection.

representations. Recently, He et al. (2022) show the effectiveness of target-related Wikipedia knowledge for classification-based stance detection. We also follow the idea and incorporate target-related Wikipedia knowledge for our generation model.

We evaluate our method on VAST. Experimental results show that the conditional generation formulation can achieve better performance compared to classification, demonstrating the effectiveness of connecting input, target, and label semantics for stance detection. Further analysis illustrates the benefits of joint target prediction, unlikelihood training, and Wikipedia knowledge. Our model can achieve new state-of-the-art performance, outperforming several strong baselines from previous work.

2 Approach

In this section, we will discuss our approach to zero-shot and few-shot stance detection. We will first introduce the problem formulation, and then discuss our generation-based framework.

2.1 Problem Formulation

Stance detection aims to identify the polarity of an input text with regard to a specific target. Formally, a sample instance can be considered as a triple $(\boldsymbol{x}, \boldsymbol{t}, y)$, where $\boldsymbol{x}$ and $\boldsymbol{t}$ are two sequences of tokens, representing input text and target respectively. $y \in \{\text{supportive (pro)}, \text{opposite (con)}, \text{neutral}\}$ represents then stance label.

A stance-detection model is to infer the stance label y given $\boldsymbol{x}$ and $\boldsymbol{t}$ with parameter θ:

$$f(\boldsymbol{x}, \boldsymbol{t}; \theta) = y.$$

In the zero-shot and few-shot stance detection dataset with varied targets (Allaway and McKeown, 2020), many target tokens only occur zero or a few times in the training set.

2.2 A Generation-Based Framework

Generation-based frameworks have demonstrated their effectiveness for problems beyond traditional generation tasks (Lewis and Fan, 2019; Yan et al., 2021; Li et al., 2021; Raffel et al., 2022). We use a conditional generation model for this problem, where the condition is a partially-filled template with the input text. The template is two sentences describing the target and stance with a `<stance>` placeholder for stance detection. An example of the partially-filled template with input text and output is shown in Figure 1.

Our base model is BART (Lewis et al., 2020), an encoder-decoder language model pretrained with denoising objectives, which is similar to our generation-based formulation. The generation process can be considered as using the conditional probability to select a new token at each step given input and previously generated tokens:

$$p(\boldsymbol{o} \mid g(\boldsymbol{x}, \boldsymbol{t}); \theta) = \prod_{i=1}^{|\boldsymbol{o}|} p(o_i \mid \boldsymbol{o}_{<i}, g(\boldsymbol{x}, \boldsymbol{t}); \theta),$$

where $g(\boldsymbol{x}, \boldsymbol{t})$ represents the transformation function that fills the target $\boldsymbol{t}$ into the template and forms the input sequence with the input text $\boldsymbol{x}$. Specifically, $g(\boldsymbol{x}, \boldsymbol{t})$ will generate a combination of input text and template with special tokens: "`<s>` template `</s></s>` $\boldsymbol{x}$ `</s>`". The template contains two sentences: "The target is `<target>`. The stance is `<stance>`". We will fill in `<target>` placeholder with the actual target and keep the `<stance>` placeholder for the decoder to generate.

The generated output $\boldsymbol{o}$ is a fully-filled template, where both target and stance placeholders are replaced by actual or predicted values. The model is trained by minimizing the log-likelihood over the whole generated sequence:

$$\begin{aligned}\mathcal{L}_s &= -\log p(\boldsymbol{o} \mid g(\boldsymbol{x}, \boldsymbol{t}); \theta) \\ &= -\sum_{i=1}^{|\boldsymbol{O}|} \log p(o_i \mid \boldsymbol{o}_{<i}, g(\boldsymbol{x}, \boldsymbol{t}); \theta).\end{aligned}$$

The final predicted stance label is obtained with a post-processing function that tries to find the polarity word after the prompt for stance.

2.2.1 Joint Target Prediction

Another advantage of using generation-based architecture is that we can leverage auxiliary generative

Stance Detection	
Input	Target is high-speed rail. Stance is `<stance>`.
Output	Target is high-speed rail. Stance is supportive.
Target Prediction	
Input	Stance is supportive. Target is `<target>`.
Output	Stance is supportive. Target is high-speed rail.
Unlikelihood Training	
Input	Target is high-speed rail. Stance is `<stance>`.
Output	Target is high-speed rail. Stance is opposite.

Table 2: Examples input and output templates for stance detection, target prediction, and unlikelihood training.

tasks to help train stance detection. We use target prediction, which is to infer the target tokens $\boldsymbol{t}$ given stance label y and input text $\boldsymbol{x}$:

$$f_t(\boldsymbol{x}, y; \theta) = \boldsymbol{t}.$$

Target prediction can provide the connection of stance to target in an opposite direction of stance detection. It can also enhance the representation of target tokens by learning to decode them.

The input sequence of target prediction is similar to stance detection, consisting of a partially-filled template and input text. The template used for joint target prediction is slightly different than the one used for stance detection, where we switch the position of two sentences so that the stance information shows up first. We will fill in the actual stance text in the input sequence, and leave the `<target>` placeholder for the decoder to generate.

2.2.2 Unlikelihood Training

Log-likelihood objective optimizes the likelihood over the entire distribution. However, in our task, especially when generating the stance labels, we should specifically focus on several candidate tokens. Therefore, we introduce unlikelihood training (Welleck et al., 2020), where we use unlikely tokens, *i.e.* incorrect stance predictions, to replace the ground-truth sequence and optimize with the unlikelihood loss for the replaced tokens.

Specifically, for an output sequence $\boldsymbol{o}$, we assume o_k is the stance label and replaced it with an incorrect stance prediction o'_k while keeping other tokens to form incorrect sequence $\boldsymbol{o}'$. The combination of likelihood and unlikelihood will be:

$$\begin{aligned}\mathcal{L}_u = \quad & \log p\left(o'_k \mid \boldsymbol{o}'_{<k}, g\left(\boldsymbol{x}, \boldsymbol{t}\right); \theta\right) \\ & - \sum_{i \neq k} \log p\left(o'_i \mid \boldsymbol{o}'_{<i}, g\left(\boldsymbol{x}, \boldsymbol{t}\right); \theta\right),\end{aligned}$$

For each ground-truth sequence, we can construct two sequences for unlikelihood training with the other two incorrect stance labels. Table 2 illustrates the examples for different input and output templates for stance prediction, target prediction, and unlikelihood training.

2.2.3 Incorporating Wikipedia Knowledge

He et al. (2022) collect relevant Wikipedia snippets for each target and propose to incorporate Wikipedia knowledge to enhance target representations for BERT-based (Devlin et al., 2019) classification, which demonstrates a significant improvement. We follow He et al. (2022) and incorporate Wikipedia knowledge into our generation-based method. Specifically, we append Wikipedia snippets to the end of our input sequence: "`<s>` template `</s></s>` $\boldsymbol{x}$ `</s></s>` Wikipedia snippet `</s>`". We use the new input sequence to perform both training and inference while the output sequences remain as the fully-filled templates.

2.2.4 Training Objective

The final training objective is the combination of loss functions from stance detection, target prediction, and unlikelihood training:

$$\mathcal{L} = \mathcal{L}_s + \alpha_t \mathcal{L}_t + \alpha_u \mathcal{L}_u,$$

where $\mathcal{L}_t$ represents the log-likelihood loss over the output template for target prediction, α_t, α_u are used to balance different loss functions.

3 Experiments

3.1 Data

VAST contains 18,548 examples from *New York Times* "Room for Debate" section with 5,630 different targets for zero-shot and few-shot stance detection. The original examples of VAST are collected from Habernal et al. (2018) under Apache-2.0 license[2]. We use Wikipedia knowledge collected by He et al. (2022), which uses API to crawl Wikipedia pages for targets. Wikipedia content can be used under Creative Commons Attribution Share-Alike license (CC-BY-SA)[3]. We use the same training/devlopment/test split as Allaway and McKeown (2020).

3.2 Experimental Setup

We conduct our experiments on VAST (Allaway and McKeown, 2020). We compare our model

[2] https://github.com/UKPLab/argument-reasoning-comprehension-task/blob/master/LICENSE

[3] https://en.wikipedia.org/wiki/Wikipedia:Reusing_Wikipedia_content

Model	Precision	Recall	F_1
BERT Classification	72.6	72.0	72.1
BART w/ Template	75.7	75.1	75.3
+ Topic Prediction	76.0	75.6	75.7
+ Unlikelihood	76.4	75.9	75.9
+ Wikipedia	**78.0**	**77.3**	**77.4**

Table 3: Performance of different model variants on the overall precision, recall and F_1 on the development set (%). Each of our model variants is on top of the variant from its previous row.

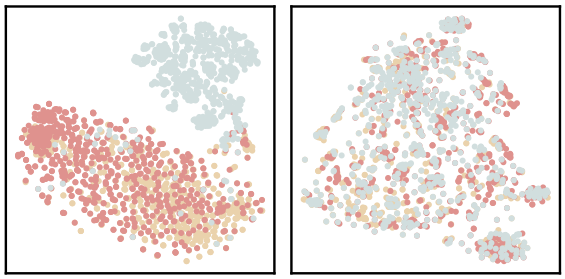
Model	Zero-Shot	Few-Shot	Overall
TGA-Net	66.6	66.3	66.5
BERT-GCN	68.6	69.7	69.2
CKE-Net	70.2	70.1	70.1
WS-BERT	75.3	73.6	74.5
Our Model	**76.4**	**78.0**	**77.3**

Table 4: Stance detection performance (%) on VAST. Our model significantly outperforms previous work on all metrics. Our results are obtained from averaging performances over 5 random seeds. $p < 0.001$ on overall F_1 using Z-test with variance as the standard deviation over multiple runs.

with several existing systems including 1) TGA-Net (Allaway and McKeown, 2020); 2) BERT-GCN (Lin et al., 2021); 3) CKE-Net (Liu et al., 2021); 4) WS-BERT (He et al., 2022). Following their setup, we use macro-average F_1 as the evaluation metric, and we report performance on the subset of test set for zero-shot and few-shot, and the overall test set.

We use BART-base[4] as our base model, of which the number of parameters is roughly consistent with baselines on BERT-base[5]. Our best model is optimized with AdamW (Loshchilov and Hutter, 2019) for 30 epochs with a learning rate of 1e-5. We use a linear scheduler with a warmup proportion of 0.1 and the training batch size is 32. We use greedy search during inference. We reported performances on development set and test set using the averaged results from 5 different random seeds. Test results are reported based on the best overall F_1 performance on the development set. α_t is set to 1 and α_u is set to 0.5. Our final model takes about 5 hours for training on one Nvidia RTX 3090 GPU.

[4] https://huggingface.co/facebook/bart-base
[5] https://huggingface.co/bert-base-uncased



(a) Our model (b) BERT classification

Figure 2: The t-SNE visualization of intermediate representations from our model and BERT classification model. Color map: Supportive, Opposite, Neutral.

3.3 Results

3.3.1 Comparing with Model Variants

We first conduct comparison of some of our model variants to illustrate the effectiveness of our proposed components. The results are shown in Table 3. From the comparison of BERT-based classification (BERT Classification) and BART-based denoising generation from templates (BART w/ Template), we can find that adopting the generation framework can significantly improve the model performance. Our proposed topic prediction and unlikelihood training can further boost performance. The final model with knowledge from Wikipedia, verifies the effectiveness of Wikipedia knowledge for stance detection with a generative framework.

3.3.2 Comparing with Existing Systems

Our overall performance is shown in Table 4. Our method can significantly outperform those previous baselines, indicating the effectiveness of our proposed generation framework for zero-shot and few-shot stance detection with varies topics.

3.4 Qualitative Analysis

Figure 2 show the t-SNE (van der Maaten and Hinton, 2008) visualization of intermediate representations before the classification layer from our model and BERT classification model on the development set. We use random initialization with perplexity as 50 for visualization and we color each visualized instance with its corresponding stance label. The visualization of BERT classification shows small clusters with hybrid labels, While we can see that instances with our generation method are clustered with labels, where neutral labels are at the top and supportive labels are generally at the bottom.

4 Related Work

Zero-shot and few-shot stance detection. Zero-shot and few-shot stance detection focus on detecting stances for unseen or low-resource targets. Allaway and McKeown (2020) construct a dataset with varied topics that can be used to test stance detection under zero-shot and few-shot settings. Previous efforts mostly focus on modeling targets, documents, or their connections. Allaway and McKeown (2020) obtain generalized topic representation through clustering. Liu et al. (2021) use commonsense knowledge graph to enhance the connection between target and document. Liang et al. (2022a,b) use contrastive learning to learn target features. He et al. (2022) incorporate Wikipedia knowledge to enhance target representations. While in our work, we use a conditional generation framework to build the connections between input, target, and label text semantics.

Text processing via conditional generation. Our work is also motivated by the recent success of tackling text processing problems as conditional generation (Lewis et al., 2020; Raffel et al., 2022). In addition to the conventional text generation problems, conditional generation frameworks are effectively applied in information extraction (Li et al., 2021), question answering (Lewis and Fan, 2019; Raffel et al., 2022) and sentiment analysis (Yan et al., 2021). In our work, we further explore stance detection via conditional generation.

5 Conclusion

In this paper, we propose a generation-based framework for zero-shot and few-shot stance detection that generate stance label from pre-defined templates. We further propose an auxiliary task, joint target prediction that takes stance label and input text to generate targets, and unlikelihood training on manually constructed incorrect generation output. Combining with Wikipedia knowledge for target from He et al. (2022), our model can achieve new state-of-the-art performance on VAST.

Limitations

Because of the nature of our framework design, our work requires a diverse set of targets during training, which is important for target prediction and therefore the stance detection method. It is difficult to be applied to other stance detection datasets when there are limited training resources with regard to targets, such as Conforti et al. (2020) and Mohammad et al. (2016). Besides, the model is trained on news-related debate corpus, so it may need further domain adaptation if applying the model to other domains such as social media.

We are using an auto-regressive generation framework, which will also require extra inference time to generate the whole output sequence compared to the classification model. We would encourage readers to compare it with classification methods for efficiency when it will be applied in a time-sensitive scenario.

References

Emily Allaway and Kathleen McKeown. 2020. Zero-Shot Stance Detection: A Dataset and Model using Generalized Topic Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8913–8931, Online. Association for Computational Linguistics.

Emily Allaway, Malavika Srikanth, and Kathleen McKeown. 2021. Adversarial learning for zero-shot stance detection on social media. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4756–4767, Online. Association for Computational Linguistics.

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885, Austin, Texas. Association for Computational Linguistics.

Costanza Conforti, Jakob Berndt, Mohammad Taher Pilehvar, Chryssi Giannitsarou, Flavio Toxvaerd, and Nigel Collier. 2020. Will-they-won't-they: A very large dataset for stance detection on Twitter. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1724, Online. 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.

Eduardo Graells-Garrido, Ricardo Baeza-Yates, and Mounia Lalmas. 2020. Representativeness of abortion legislation debate on twitter: A case study in

argentina and chile. In *Companion Proceedings of the Web Conference 2020*, WWW '20, page 765–774, New York, NY, USA. Association for Computing Machinery.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940, New Orleans, Louisiana. Association for Computational Linguistics.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A retrospective analysis of the fake news challenge stance-detection task. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1859–1874, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zihao He, Negar Mokhberian, and Kristina Lerman. 2022. Infusing knowledge from Wikipedia to enhance stance detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 71–77, Dublin, Ireland. Association for Computational Linguistics.

Myungha Jang and James Allan. 2018. Explaining controversy on social media via stance summarization. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1221–1224, New York, NY, USA. Association for Computing Machinery.

Yan Jiang, Jinhua Gao, Huawei Shen, and Xueqi Cheng. 2022. Few-shot stance detection via target-aware prompt distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 837–847, New York, NY, USA. Association for Computing Machinery.

Mike Lewis and Angela Fan. 2019. Generative question answering: Learning to answer the whole question. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Bin Liang, Zixiao Chen, Lin Gui, Yulan He, Min Yang, and Ruifeng Xu. 2022a. Zero-shot stance detection via contrastive learning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 2738–2747, New York, NY, USA. Association for Computing Machinery.

Bin Liang, Yonghao Fu, Lin Gui, Min Yang, Jiachen Du, Yulan He, and Ruifeng Xu. 2021. Target-adaptive graph for cross-target stance detection. In *Proceedings of the Web Conference 2021*, WWW '21, page 3453–3464, New York, NY, USA. Association for Computing Machinery.

Bin Liang, Qinglin Zhu, Xiang Li, Min Yang, Lin Gui, Yulan He, and Ruifeng Xu. 2022b. JointCL: A joint contrastive learning framework for zero-shot stance detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive text classification by combining GNN and BERT. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1456–1462, Online. Association for Computational Linguistics.

Rui Liu, Zheng Lin, Yutong Tan, and Weiping Wang. 2021. Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3152–3157, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. Association for Computational Linguistics.

Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic stance detection using end-to-end memory networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 767–776, New Orleans, Louisiana. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 551–557, Valencia, Spain. Association for Computational Linguistics.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chang Xu, Cécile Paris, Surya Nepal, and Ross Sparks. 2018. Cross-target stance classification with self-attention networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 778–783, Melbourne, Australia. Association for Computational Linguistics.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2416–2429, Online. Association for Computational Linguistics.

Rong Zhang, Qifei Zhou, Bo An, Weiping Li, Tong Mo, and Bo Wu. 2020. Enhancing neural models with vulnerability via adversarial attack. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1133–1146, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shaodian Zhang, Lin Qiu, Frank Chen, Weinan Zhang, Yong Yu, and Noémie Elhadad. 2017. We make choices we think are going to save us: Debate and stance identification for online breast cancer cam discussions. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 1073–1081, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

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

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

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

B ☑ Did you use or create scientific artifacts?

Introduction, Section 3.1 Data

☑ B1. Did you cite the creators of artifacts you used?
Introduction, Section 3.1 Data

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3.1 Data

☑ 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)?
Section 3.1 Data, Section 3.2 Experimental Setup

☒ 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?
We use an existing resource and detail of the data is discussed and introduced in their own published paper.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We use an existing resource and detail of the data is discussed and introduced in their own published paper.

☑ 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.
Section 3.1 Data

C ☑ Did you run computational experiments?

Left blank.

☑ 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 3.2 Experimental Setup

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 3.2 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?
Section 3.2 Experimental Setup, Table 1

☑ 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 3.2 Experimental Setup

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.