A New Direction in Stance Detection: Target-Stance Extraction in the Wild

Yingjie Li* **Krishna Garg*** **Cornelia Caragea**

University of Illinois at Chicago
{yli300,kgarg8,cornelia}@uic.edu

Abstract

Stance detection aims to detect the stance toward a corresponding target. Existing works have achieved promising progress on stance detection tasks in which the goal is to predict the stance given both a target and a text. However, they all work under the assumption that the target is known in advance, which is often not the case in the wild. Given a text from social media platforms, the target information is often unknown due to implicit mentions in the source text and it is infeasible to have manual target annotations at a large scale. Therefore, in this paper, we propose a new task Target-Stance Extraction (TSE) that aims to extract the (*target*, *stance*) pair from the text. We benchmark the task by proposing a two-stage framework that first identifies the relevant target in the text and then detects the stance given the predicted target and text. Specifically, we first propose two different settings: Target Classification and Target Generation, to identify the potential target from a given text. Then we propose a multi-task approach that takes target prediction as the auxiliary task to detect the stance toward the predicted target. We evaluate the proposed framework on both in-target stance detection in which the test target is always seen in the training stage and zero-shot stance detection that needs to detect the stance for the unseen target during the inference stage. The new TSE task can facilitate future research in the field of stance detection. We publicly release our code.[1]

1 Introduction

Stance detection aims to automatically identify people's attitude/viewpoint (e.g., *Favor* or *Against*) expressed in texts toward a target that is generally a controversial topic or political figure (ALDayel and Magdy, 2021; Küçük and Can, 2020; Hardalov et al., 2021). For example, the tweet in Figure 1 expresses a stance of "*Against*" toward the target "*Atheism*."

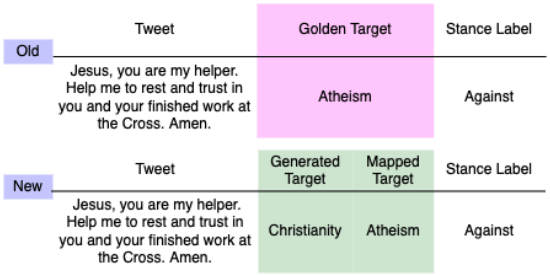


Figure 1: The comparison between the proposed Target-Stance Extraction (TSE) task and original stance detection task.

Social media platforms like Twitter, Facebook and other debate forums have become an integral way of opinion dissemination these days (Khan et al., 2021). The peculiar characteristics of these platforms are that the information is usually scattered across texts and the opinionated text could be expressed toward target entities in an implicit way. Existing methods have achieved promising performance on in-target stance detection in which same targets are seen in both train and test sets (Mohammad et al., 2016a; Sobhani et al., 2017; Li and Caragea, 2019, 2021a) and cross-target stance detection that aims to transfer the knowledge from a source target to a destination target (Augenstein et al., 2016; Xu et al., 2018; Zhang et al., 2020). However, almost all previous methods work under the assumption that the target is known or manually identified, which is often not the case in the wild. In practice, the target is unknown given a text and it is usually implicitly mentioned in the text, as can be seen from the example shown in Figure 1. Therefore, instead of detecting the stance given both the target and text, we propose a more challenging task Target-Stance Extraction (TSE) in the context of stance detection that aims to extract the (*target*, *stance*) pair from the text. The new TSE

*Both authors contributed equally to this research.
[1]https://github.com/chuchun8/TSE

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10071–10085
July 9-14, 2023

task is more challenging because it includes both target identification and stance detection.

To tackle this task, we propose a two-step framework that first identifies the relevant target in the text and then detects the stance given the predicted target and the text, as shown in Figure 1. In the first stage, we propose two different settings to identify the target discussed in a text: (1) Target Classification, where we train a text classifier (Schuster and Paliwal, 1997; Devlin et al., 2019; Nguyen et al., 2020) to predict the target as one of the pre-defined targets, and (2) Target Generation, where we leverage BART (Lewis et al., 2020) model that is pre-trained on a keyphrase generation dataset (Xiong et al., 2019; Gallina et al., 2019; Garg et al., 2022) to generate keyphrases (e.g., "*Christianity*" in Figure 1), and then map them to one of the pre-defined targets (e.g., "*Atheism*"). In the second stage, we propose a multi-task framework that takes the target prediction as the auxiliary task for stance detection. We expect the stance detection model to better capture the target-related features and to develop a better understanding of the text itself with the auxiliary task.

Our proposed two-step framework can not only be applied to in-target stance detection, but also zero-shot stance detection in which targets of test examples are not seen in the train set. We evaluate the proposed framework on the combined set of four stance datasets (Mohammad et al., 2016a; Stab et al., 2018; Glandt et al., 2021; Li et al., 2021a) for in-target stance detection. Further, we extend our framework to zero-shot stance detection and test it on six targets of diverse domains (Somasundaran and Wiebe, 2010; Mohammad et al., 2016a; Conforti et al., 2020; Miao et al., 2020; Gautam et al., 2020). It is worth noting that our primary goal is not to present a new state-of-the-art model, but to deliver a new and more challenging task to stimulate research on stance detection.

We summarize our contributions as follows:

- We propose a new Target-Stance Extraction (TSE) task, aimed to extract the pair of target and stance from each sentence.
- We benchmark the task by proposing a two-step framework that can be applied to both in-target and zero-shot stance detection.
- We propose a multi-task framework that uses the target prediction as an auxiliary task to improve the performance of stance detection.

2 Related Work

Stance Detection The stance detection task aims to detect the stance toward a specific target (Mohammad et al., 2016a; Schiller et al., 2021; Hardalov et al., 2022). The target could be defined in a variety of forms: a controversial figure (Darwish et al., 2017; Grimminger and Klinger, 2021; Li et al., 2021a), a hot topic such as gun control (Hasan and Ng, 2014; Mohammad et al., 2016a; Stab et al., 2018; Vamvas and Sennrich, 2020; Conforti et al., 2020; Glandt et al., 2021) or a claim (Rao and Pomerleau, 2017; Derczynski et al., 2017; Gorrell et al., 2019). In previous works, the target is usually manually provided along with the input sentence to a stance classifier. However, given a post on social media, we may not have a direct clue about the target information due to their implicit mentions, and it is infeasible to do large-scale target annotations by humans. Motivated by this observation, we propose a new task named Target-Stance Extraction (TSE) that aims to extract both the target and the corresponding stance from a given text.

Besides the in-target stance detection (Mohammad et al., 2016a; Li and Caragea, 2021b) in which the test target is seen in the training stage, cross-target stance detection (Augenstein et al., 2016; Xu et al., 2018; Zhang et al., 2020; Liang et al., 2021) and zero-shot stance detection (Allaway and McKeown, 2020; Liang et al., 2022; Li et al., 2023) have also drawn a lot of attention recently. In cross-target stance detection, a classifier is adapted from a different but related target to a destination target in a one-to-one way, whereas in zero-shot stance detection we need to detect the stance for a variety of unseen targets at the inference stage. In this paper, we evaluate our proposed framework in both in-target and zero-shot settings.

Keyphrase Generation / Extraction Keyphrase generation or extraction is the task where given a source document (e.g., a scientific article, newspaper article, or webpage), we predict the keyphrases that best describe or summarize that document (Garg et al., 2022; Ray Chowdhury et al., 2022, 2019; Alzaidy et al., 2019; Patel and Caragea, 2019; Meng et al., 2017; Yuan et al., 2020; Ye et al., 2021; Florescu and Caragea, 2017; Sterckx et al., 2016; Caragea et al., 2014). In the context of stance detection, we can use keyphrase generation models to generate keyphrases that are target-related words give an input text. To our knowledge, target-related

keyphrase generation task has not been explored before for stance detection.

The most popular paradigm for the keyphrase generation task is the One2Seq encoder-decoder framework (Meng et al., 2017) where given a document, we generate a sequence of *[SEP]* separated keyphrases in an auto-regressive way. We use the pre-trained BART model (Lewis et al., 2020) fine-tuned separately on three keyphrase generation datasets, i.e., OpenKP (Xiong et al., 2019), KP-Times (Gallina et al., 2019), and FullTextKP (Garg et al., 2022) and generate keyphrases using the One2Seq model.

3 Task and Datasets

3.1 Task Definition

Let $D_{tr} = \{x_i, t_i, y_i\}_{i=1}^{n}$ be a train set where x_i is a sequence of words, t_i is the target holding the stance and y_i is the stance label. In the original stance detection task the aim was to only detect the stance y_i given the target t_i and the text x_i.

Target-Stance Extraction Objective In our proposed Target-Stance Extraction (TSE) task the goal is to extract the target-stance pair (t_i, y_i) given x_i.

3.2 Datasets

In-Target TSE For in-target TSE, we conduct experiments on the merged set of four stance detection datasets to evaluate the proposed framework. 1) **SemEval-2016** (SE) (Mohammad et al., 2016b) contains 5 pre-defined targets, including *Atheism*, *Climate Change is a Real Concern*, *Feminist Movement*, *Hillary Clinton* and *Legalization of Abortion*. Each sample is annotated with *Favor*, *Against* or *None*. 2) **AM** (Stab et al., 2018) is an argument mining dataset containing 8 targets, including *Abortion*, *Cloning*, *Death Penalty*, *Gun Control*, *Marijuana Legalization*, *Minimum Wage*, *Nuclear Energy* and *School Uniforms*. Each sample is annotated with *Support*, *Oppose* or *Neutral*. 3) **COVID-19** (C19) (Glandt et al., 2021) contains 4 targets related to COVID-19: *Wearing a Face Mask*, *Anthony S. Fauci*, *School Closures* and *Stay at Home Orders*. Each sample can be classified as *Favor*, *Against* or *None*. 4) **P-Stance** (PS) (Li et al., 2021a) contains 3 targets related to the 2020 U.S. presidential election: *Donald Trump*, *Joe Biden* and *Bernie Sanders*. Each instance is annotated with *Favor* or *Against*.

Train, validation and test sets are provided for the AM, COVID-19, and P-Stance datasets. For SemEval-2016, train and test sets are provided and we split the train set into train and validation sets. We remove the target *Climate Change* of SemEval-2016 from training for the usage of zero-shot setting. Data statistics and examples of these datasets are shown in Tables 1 and 2.

Zero-Shot TSE We also curate a new zero-shot dataset from existing datasets to test the model performance on unseen targets during the inference stage. We collect 500 samples for each of the following targets from its original dataset: 1) *Creationism* (Somasundaran and Wiebe, 2010), 2) *Gay Rights* (Somasundaran and Wiebe, 2010), 3) *Climate Change is a Concern* (Mohammad et al., 2016a), 4) *MeToo Movement* (Gautam et al., 2020), 5) *Merger of Disney and Fox* (Conforti et al., 2020), 6) *Lockdown in New York State* (Miao et al., 2020).

To mimic the real-world scenario that a text may contain no targets of interest, we consider an additional target label *Unrelated* in both in-target and zero-shot settings. We provide the details about the curation of such samples in the Appendix A. We maintain a ratio of 5:1 for interested targets vs. the *Unrelated* category in the final datasets for both in-target and zero-shot TSE. The numbers of targets for in-target and zero-shot datasets are 18[2] and 6, respectively, and we add the *Unrelated* category in each dataset.

4 Approach

As discussed in the previous section, TSE is a challenging task that involves both target identification and stance detection given a text. To tackle this task, we propose a two-stage framework, in which we first identify the target from a given text using either a target classification or target generation approach and then detect the stance toward the predicted target with a stance classifier in the second stage. The overall framework of our proposed approach is shown in Figure 2.

4.1 Stage 1: Target Identification

In this stage, we extract the target from the text based on either training classifiers, e.g., BiLSTM or BERT, to predict the target from a set of pre-defined targets or by using a BART-fine-tuned keyphrase generation module to generate keyphrases for the text and then map them to the pre-defined set of

[2]We merge the semantically similar targets *Abortion* (AM) and *Legalization of Abortion* (SemEval-2016) for the merged training dataset.

Dataset	#Train	#Val	#Test	Targets
SemEval-2016	2,160	359	1,080	Atheism, Feminist Movement, Hillary Clinton, Legalization of Abortion
AM	18,341	2,042	5,109	Abortion, Cloning, Death Penalty, Gun Control, Marijuana Legalization, Minimum Wage, Nuclear Energy, School Uniforms
COVID-19	4,533	800	800	Face Masks, Fauci, Stay at Home Orders, School Closures
P-Stance	17,224	2,193	2,157	Joe Biden, Bernie Sanders, Donald Trump
Zero-Shot	-	-	3,000	Creationism, Gay Rights, Climate Change is a Concern, MeToo Movement, Merger of Disney and Fox, Lockdown in New York State

Table 1: Data split statistics for SemEval-2016, AM, COVID-19, P-Stance and Zero-Shot datasets.

Dataset	Target	Tweet	Stance
SemEval-2016	Atheism	Religious leaders are like political leaders - they say what they think people want to hear. #freethinker #SemST	Favor
AM	Gun Control	Restrictions on gun ownership will only encourage outlaws to have heavy ammunition and high calibre weapons.	Against
COVID-19	Face Masks	@MrMasonMills @YcmiYcmiu There is air in houses/buildings too. Are we expected to live in a mask constantly?	Against
P-Stance	Donald Trump	There was no collusion Collusion is not a crime Even if it's a crime, it's doesn't matter. It's ALL HILLARY AND OBAMA'S FAULT The evolution of the #Trump defense	Favor
Zero-Shot	Gay Rights	Yes! You rock gay people. They are people just like we are and if two men want to marry each other, than go for it	Favor

Table 2: Examples from stance detection datasets.

targets. Our intuition is that the keyphrases corresponding to a text capture its essence and they should correlate well with the target towards which the stance is expressed. For instance, in Figure 1, the generated target *Christianity* quite succinctly captures the essence from the tweet *Jesus, you are my helper...* and at the same time, the generated target *Christianity* correlates semantically well to the golden target *Atheism*.

Target Classification In this approach, we train a classifier based on the merged dataset with texts as inputs and their corresponding targets as the ground truth labels. Note that the stance labels are *not used* in this target classification task. We discuss this approach in more details in §5.2.

Target Generation In this approach, we first fine-tune a BART model on one of the keyphrase generation datasets separately,[3] i.e., OpenKP (Xiong et al., 2019), KPTimes (Gallina et al., 2019) and FullTextKP (Garg et al., 2022). The BART keyphrase generation model is used to generate keyphrases (e.g., "*Christianity*") given a text. Note that the generated keyphrases may not directly belong to any of the target classes we are interested in. Therefore, a similarity mapping is adopted to map the generated keyphrases into one of the pre-defined targets.

For similarity mapping, we first train a FastText model (Bojanowski et al., 2017) on the train set of the merged dataset. Our choice for FastText is motivated by its efficiency while maintaining comparative performance with BERT-based models. Then we obtain word embeddings of the generated keyphrases by sending them as inputs to the FastText model. Finally, a cosine similarity score is calculated between the embeddings of generated keyphrase and each pre-defined target. We predict the target that has the highest similarity score with the generated keyphrase. Note that the generated keyphrase is classified as *Unrelated* if the highest similarity score is below a specific threshold.

4.2 Stage 2: Stance Detection

Given a text in the wild, the target information is usually unknown, and thus we first predict the target from either target classification or target generation in the first stage. Then in the second stage, we use a stance classifier that is trained on the merged set to detect the stance of predicted targets.

For stance detection, we train a stance classifier as follows. Given a text x_i and a target t_i, we first formulate the input as a sequence s_i = [$[CLS]$ t_i

[3]We also fine-tuned the BART model on stance datasets to directly learn to generate the targets of interest. However, it shows much worse performance than the models trained on keyphrase generation datasets potentially due to the smaller size of the stance datasets.

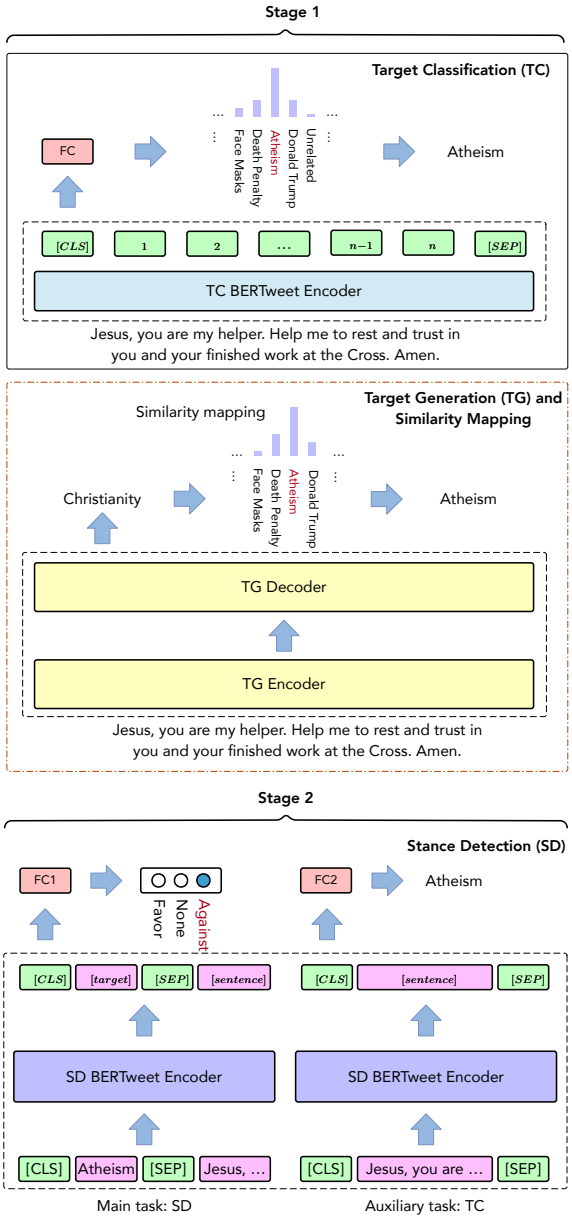


Figure 2: Model architecture of our two-stage approach for Target-Stance Extraction task. Models in black dash boxes can be replaced with other baselines. Model architecture in the red dash box indicates the alternative solution in the first stage.

$[SEP]$ $x_i]$ where $[CLS]$ is a token that encodes the sentence and $[SEP]$ is used to separate the sentence x_i and the target t_i. Then the representation of $[CLS]$ token is used to predict the stance toward target t_i. Note that t_i is the golden target in the training stage and is the predicted target from target identification at the inference stage.

To facilitate a model's ability to capture target-related features that are of vital importance to stance detection, we propose a multi-task framework that uses target prediction as the auxiliary task that aims to predict the target given the input text for stance detection. More specifically, in the auxiliary task, we formulate the input as $[[CLS]$ x_i $[SEP]]$ and the golden label is target t_i. The layers of encoders are shared across tasks and each task has its specific fully-connected layer on top, which is updated during the training. We expect the model to be able to put more attention on target-related words with the auxiliary task, and thus show better performance on stance detection task. The overall architecture is shown in Figure 2.

Note that the auxiliary task is similar with target classification of Stage 1 and thus it cannot be used in zero-shot stance detection. In zero-shot setting, we first leverage the keyphrase generation model for target prediction and then detect the stance toward the predicted target with the multi-task stance model. In order to be consistent with the target generation setting that decouples target identification from stance detection, we train a separate target classification model (BERTweet or BiLSTM) in Stage 1 and a multi-task model (BERTweet or other stance detection baselines) in Stage 2 for stance detection. However, note that the target classification of the auxiliary task can be used for the in-target TSE setting.

5 Experimental Settings

5.1 Evaluation Metrics

Target-Stance Extraction Target-Stance Extraction (TSE) task aims to extract the target-stance pair from a given text. We propose to solve this task by first identifying the target from the text and then detecting the stance toward the predicted target. We gather the (*predicted target*, *predicted stance*) pair for evaluation. For TSE task, we use the F_1 and accuracy as the evaluation metrics. The calculation of F_1 is shown as follows:

$$Precision = \frac{\#correct}{\#predict}, \quad (1)$$

$$Recall = \frac{\#correct}{\#gold}, \quad (2)$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

where $\#correct$ denotes the number of target-stance pairs correctly predicted by the model, $\#predict$ denotes the number of target-stance pairs whose target is predicted as one of our interested

targets (not *Unrelated*) by the model, $\#gold$ denotes the number of target-stance pairs whose target is not *Unrelated* in the dataset.

For accuracy, a prediction pair can be counted as a correct prediction if it satisfies one of the following two conditions: 1) the predicted target-stance pair is the same as the golden one if the golden target is not *Unrelated*, 2) the predicted target and the golden target are both *Unrelated*. Since we show no interest in *Unrelated* category, we do not detect the stance toward the *Unrelated* category.

Target Identification We evaluate the target classification and target generation using micro-averaged F_1 over the golden targets in each dataset.

Stance Detection For the original formulation of the stance detection task, we use the F_{avg}, macro-average of F_1 (F_{mac}) and micro-average of F_1 (F_{mic}) as the evaluation metrics following the previous work (Mohammad et al., 2016b). F_{avg} is calculated as the average F_1 of *Favor* and *Against* toward each dataset. Further, F_{mac} is calculated by averaging the F_{avg} across all four datasets. We obtain F_{mic} by averaging the F_1 of *Favor* and *Against* across the merged dataset.

5.2 Baseline Models

Target Classification As discussed in §4.1, this task involves training a classifier which can predict the target mentioned in the given tweet. We use the following neural network based classifiers:

- **BiLSTM** (Schuster and Paliwal, 1997): We use BiLSTM networks followed by two linear layers to predict the target given a text.
- **BERT** (Devlin et al., 2019): A pre-trained language model that predicts the target by appending a linear layer to the hidden representation of $[CLS]$ token. We fine-tune the BERT-base on the target classification task.
- **BERTweet** (Nguyen et al., 2020): This variant of BERT is pre-trained on 845M English Tweets following the training procedure of RoBERTa (Liu et al., 2019). We fine-tune the BERTweet on the target classification task.

Target Generation As discussed in §4.1, we train the BART model separately on the keyphrase generation datasets as described below:

- **BART-OpenKP**: BART, pre-trained on the OpenKeyPhrase (OpenKP) dataset (Xiong et al., 2019), is used as a baseline for generating keyphrases for the input texts. OpenKP is a large-scale open domain keyphrase extraction dataset consisting of 148,124 annotated real-world webpages.
- **BART-KPTimes**: BART, pre-trained on the KPTimes (Gallina et al., 2019) dataset, serves as another baseline for target generation. KPTimes is a large-scale keyphrase generation dataset consisting of $\sim$ 280,000 news articles with the editor-curated keyphrases.
- **BART-FullTextKP**: BART is pre-trained on the FullTextKP (Garg et al., 2022) dataset. FullTextKP is a collection of 142,844 scientific articles along with the annotated keyphrases. We use the version of FullTextKP which contains only the titles and abstracts of those articles.

Stance Detection We first train the model on the merged dataset and then apply the well-trained model to predict the stance toward *the predicted target* from the target identification stage. We conduct experiments with the following baselines:

- **BiLSTM** (Schuster and Paliwal, 1997): A BiLSTM model is used to predict the stance without considering the target information.
- **BiCond** (Augenstein et al., 2016): A BiLSTM model that uses a conditional encoding method. The target is first encoded by a BiLSTM, whose hidden representations are then used to initialize another BiLSTM with sentences as inputs.
- **TAN** (Du et al., 2017): An attention-based BiLSTM model that learns the correlation between target and sentence representations.
- **CrossNet** (Xu et al., 2018): A variant of BiCond model, which adds an attention layer to capture the important words of inputs.
- **TGA-Net** (Allaway and McKeown, 2020): A BERT-based model that uses topic-grouped attention.
- **BERTweet** (Li et al., 2021a,b): A pre-trained language model that is fine-tuned by adding a linear layer to the hidden representation of the *[CLS]* token. The input is formulated as: *[CLS] target [SEP] text*.

Model	SE	AM	C19	PS	Merged
BiLSTM	52.07	54.56	50.00	60.79	61.00
BERT	77.38	70.40	66.38	70.10	74.70
BERTweet	**81.27**	**70.55**	**66.54**	**72.25**	**75.59**

Table 3: Performance comparisons of different models in micro-averaged F_1 on target classification.

Model	SE	AM	C19	PS	Merged
OpenKP	**32.22**	61.24	28.25	43.81	43.02
KPTimes	30.83	**66.31**	26.00	**63.65**	**48.31**
FullTextKP	28.06	64.67	**29.38**	44.83	43.81

Table 4: Performance comparisons of different models in micro-averaged F_1 on target generation.

6 Results and Analysis

In this section, we first present the results for target classification and target generation in §6.1. We then present the set of experiments performed on the in-target TSE task and show the results obtained by using the aforementioned baselines in §6.2. In the next section §6.3, we report the results for the zero-shot TSE task where targets of test set are not seen in the train set. Finally, we study the performance of multi-task models in §6.4. Each result is the average of three runs with different initializations.

6.1 Target Classification and Target Generation

For target classification, BERT-based models consistently outperform the BiLSTM model by a wide margin and BERTweet further supersedes BERT across all datasets, as shown in Table 3. We can also observe that all models achieve relatively low performance on the COVID-19 dataset. One reason is that targets in this dataset are all closely related to COVID-19 and thus share a lot of topics / commonalities, which make the target classification task more challenging.

For target generation, we report the performance of different pre-trained BART models in Table 4. We can observe that the overall performance of target generation is lower than the target classification task, which implies that the target generation task is more challenging. However, unlike the target classification models that can only be applied to in-target stance detection, target generation models can be directly extended to zero-shot stance detection that needs to detect the stance for targets unseen during training. In addition, the keyphrase generation models produce interesting generations as shown in Appendix B, that could be leveraged for other research purposes for stance detection such as data augmentation as part of future work.

6.2 In-Target TSE

TSE with Target Classification In Table 5, we report the performance of our proposed two-stage framework with target classification. Stance detection baselines are trained in our proposed multi-task setting on the merged dataset. Note that the BiLSTM, BERT and BERTweet in the first row of Table 5 are the target classification models. GT means that all ground-truth targets are used for stance detection (Stage 2). First, it can be seen that the overall performance of stance baselines is relatively low, which indicates that our proposed TSE task is very challenging. Second, we can observe that stance classifier BERTweet achieves the best performance across all target classification models, which is consistent with our observation in Table 8 that BERTweet performs best on in-target stance detection. Third, we can observe that each stance classifier achieves the best performance on target classifier BERTweet also due to its higher accuracy in target identification. Fourth, a significant performance drop can be seen between GT and each target classification model, which indicates that it is challenging to correctly identify the targets in our proposed framework.

TSE with Target Generation Besides target classification, we report the performance of our proposed two-stage framework with target generation in Table 6. Stance detection baselines are trained in our proposed multi-task setting on the merged dataset. The OpenKP, KPTimes, and FullTextKP of the first row indicate the train sets of the keyphrase generation models. First, we see that stance classifiers show lower performance in the target generation setting in overall than the target classification setting. One explanation is that keyphrases generated by the keyphrase generation models might be related to other topics contained in the sentence. However, in most datasets, one sentence is annotated with only one target and thus the generated keyphrases may be mapped to wrong targets.

Second, we can observe that stance classifiers achieve higher performance in evaluation metric F_1 over accuracy in Table 6, which is different from the observation in Table 5. The reason is that target

Model	BiLSTM		BERT		BERTweet		GT	
	F_1	Acc	F_1	Acc	F_1	Acc	F_1	Acc
BiLSTM	35.38	44.64	44.81	53.15	45.46	53.61	65.23	71.16
BiCond	35.36	44.63	44.94	53.26	45.59	53.72	65.61	71.48
TAN	36.69	45.73	46.32	54.41	47.02	54.91	67.33	72.90
CrossNet	36.30	45.41	45.81	53.98	46.41	54.40	67.09	72.70
TGA-Net	39.23	47.83	49.46	57.02	50.31	57.65	71.73	76.55
BERTweet	**41.35**	**49.59**	**52.24**	**59.33**	**53.30**	**60.13**	**75.28**	**79.49**

Table 5: Performance comparisons of different models in F_1 and accuracy on the merged dataset and in-target TSE task with target classification setting. GT: ground-truth targets are used for stance detection (Stage 2), which is the upper bound of model performance in our proposed framework.

Model	OpenKP		KPTimes		FullTextKP		GT	
	F_1	Acc	F_1	Acc	F_1	Acc	F_1	Acc
BiLSTM	28.40	26.50	32.69	30.06	29.24	26.86	65.23	71.16
BiCond	28.64	26.71	32.94	30.29	29.22	26.84	65.61	71.48
TAN	29.75	27.72	34.13	31.37	30.52	28.03	67.33	72.90
CrossNet	29.25	27.27	33.63	30.92	30.19	27.73	67.09	72.70
TGA-Net	31.89	29.65	36.76	33.77	32.86	30.16	71.73	76.55
BERTweet	**34.02**	**31.57**	**38.92**	**35.74**	**35.16**	**32.26**	**75.28**	**79.49**

Table 6: Performance comparisons of different models in F_1 and accuracy on the merged dataset and in-target TSE task with target generation setting. GT: ground-truth targets are used for stance detection (Stage 2), which is the upper bound of model performance in our proposed framework.

classifiers show much better performance on the class *Unrelated* because samples of *Unrelated* are seen during training. However, in target generation, we predict the generated keyphrases as *Unrelated* category with a threshold, which is not accurate in some cases and introduces another source of error.

Third, we can observe that BERTweet still achieves the best performance across all keyphrase generation models, indicating its effectiveness on in-target stance detection.

Fourth, we can see that stance classifiers generally achieve better performance with the generation model trained on KPTimes, which is consistent with our observation in Table 4.

Fifth, as before, we can observe a significant performance drop between GT and each target generation model (even higher than the target classification). This is not surprising since target generation is even more challenging than target classification.

6.3 Zero-Shot TSE

To investigate the ability of different baselines in addressing the unseen targets, we further evaluate the performance of baselines on zero-shot stance detection where targets of test set are not seen in train and validation sets. Table 7 shows performance comparisons of baseline models on the zero-shot TSE task in target generation setting. Note that target classification cannot be directly applied to identify the target in zero-shot tasks because given an input sentence, the predicted target of target classification must be one of the seen targets in train set. We can observe that zero-shot baseline TGA-Net achieves the best performance across all keyphrase generation models, indicating that TGA-Net shows better ability to generalize to unseen targets with topic-grouped attention. In addition, stance classifiers show best results with the generation model trained on KPTimes, which is consistent with the results in Table 4. It can be seen that even GT does not perform well on the zero-shot dataset, indicating the difficulty of our zero-shot task.

6.4 Effectiveness of Multi-Task Learning on Stance Detection

As mentioned before, all results reported in §6.2 and §6.3 are based on multi-task models. To investigate the effectiveness of multi-task learning, we compare the performance of multi-task models with single-task models in Table 8. Each model is trained and validated on the merged set and tested on the individual datasets where targets are golden targets instead of generated targets for a better understanding of experimental results. We can observe that all multi-task models consistently outperforms single-task models on all datasets, demon-

Model	OpenKP		KPTimes		FullTextKP		GT	
	F_1	Acc	F_1	Acc	F_1	Acc	F_1	Acc
BiLSTM	12.77	11.81	13.15	12.10	12.95	11.91	27.42	39.52
BiCond	13.60	12.57	14.31	13.17	13.77	12.66	28.98	40.81
TAN	13.30	12.31	13.29	12.23	13.53	12.44	27.51	39.59
CrossNet	14.38	13.29	14.89	13.69	14.39	13.23	30.73	42.28
TGA-Net	**21.47**	**19.76**	**22.83**	**20.95**	**21.36**	**19.61**	**40.94**	**50.79**
BERTweet	19.11	17.60	20.45	18.78	20.11	18.46	38.51	48.76

Table 7: Performance comparisons of different models in F_1 and accuracy on the zero-shot dataset and zero-shot TSE task with target generation setting. GT: ground-truth targets are used for stance detection (Stage 2), which is the upper bound of model performance in our proposed framework.

Model	SE	AM	C19	PS	$\mathbf{F_{mac}}$	$\mathbf{F_{mic}}$
Single-Task						
BiLSTM	53.05	45.70	53.34	73.62	56.43	58.75
BiCond	52.63	46.96	58.73	74.56	58.22	60.14
TAN	55.26	50.85	56.83	74.67	59.40	61.60
CrossNet	61.06	50.79	65.89	75.08	63.21	63.03
TGA-Net	63.74	58.71	64.70	77.70	66.21	67.56
BERTweet	68.03	64.31	72.99	81.47	71.70	72.26
Multi-Task						
BiLSTM	57.03	47.45	59.35	74.22	59.51	60.63
BiCond	56.22	47.11	61.69	75.29	60.08	60.98
TAN	58.54	52.13	60.31	76.29	61.82	63.32
CrossNet	61.41	51.30	67.65	76.45	64.20	63.89
TGA-Net	64.05	59.26	66.77	78.67	67.19	68.12
BERTweet	**70.62**	**64.85**	**74.42**	**81.67**	**72.89**	**73.01**

Table 8: Performance comparisons of different models on in-target stance detection. We report F_{avg}, macro-average of F_1 (F_{mac}) and micro-average of F_1 (F_{mic}).

strating the effectiveness of the multi-task learning. Specifically, the average improvements of multi-task models over single-task models are 2.35%, 0.80%, 2.95% and 0.92% in F_{avg} on SemEval-2016, AM, COVID-19, and P-Stance datasets, respectively. In addition, we can see that multi-task models achieve larger improvements on SemEval-2016 and COVID-19 datasets. One possible reason is that there are fewer train samples in SemEval-2016 and COVID-19 datasets than the rest and thus the auxiliary task of identifying targets can help models better capture the target-related features.

7 Conclusion

In this paper, we introduce a new Target-Stance Extraction (TSE) task to identify both target and corresponding stance in the wild. Different from original stance detection task that aims to only detect the stance given the target and text, our proposed task includes both target identification and stance detection, which makes it a more challenging task. We benchmark the task by proposing a two-stage framework that first identifies the target from a text and then detects the stance toward the predicted target. Our two-stage framework can not only be applied to in-target stance detection but also zero-shot stance detection. In addition, we propose a multi-task approach that takes target prediction as an auxiliary task to improve the task performance of stance detection.

It is worth noting that the primary goal of this paper is the introduction of new stance detection task. The proposed framework provides a good starting point and leaves much room for further improvements. Future work includes improving the target identification task, e.g., with a better mapping strategy.

8 Limitations

We present a novel (Target, Stance) pair Extraction task (TSE) for understanding the stance of interesting topics in the wild. There are two potential limitations to our work. First, the mapping module requires a predefined list of targets. Without the predefined list of targets, it is very difficult to understand the correctness of stance labels for the predicted targets in the absence of gold labels. On the other hand, the predefined list of targets makes the entire system end-to-end and automatically evaluable. Second, the process of mapping might become too slow if the number of targets of interest grows bigger. Future works include solving the given limitations and extracting (target, stance) pairs in a unified setting. However, the primary contribution of the work is not to present a fully robust pipeline model but to present a novel, interesting, and challenging task to the community working in stance detection.

9 Ethical Considerations

Beyond the proposed two-step framework that helps collect the stance in the wild, it is very important to consider the ethical implications of stance detection systems. Since stance detection systems could automatically collect and aggregate the topical stance for a specific target, these systems may have significant impact on decision-making. Algorithms are not perfect, and thus a potential harm is that these systems may make incorrect predictions and further mislead the decision-making. Researchers should be aware of potential harms from the misuse of stance detection systems, and should respect people's privacy during the data collection.

Acknowledgments

We thank the National Science Foundation for support from grants IIS-1912887, IIS-2107487, and ITE-2137846 which supported the research and the computation in this study. We also thank our reviewers for their insightful feedback and comments.

References

Abeer ALDayel and Walid Magdy. 2021. Stance detection on social media: State of the art and trends. *International Journal on Information Processing and Management*, 58(4).

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.

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*, WWW '19, page 2551–2557, New York, NY, USA. Association for Computing Machinery.

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.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1435–1446, Doha, Qatar. 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.

Kareem Darwish, Walid Magdy, and Tahar Zanouda. 2017. Trump vs. Hillary: What went viral during the 2016 US presidential election. In *Social Informatics*, pages 143–161.

Leon Derczynski, Kalina Bontcheva, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Arkaitz Zubiaga. 2017. SemEval-2017 task 8: RumourEval: Determining rumour veracity and support for rumours. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 69–76.

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.

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3988–3994.

Corina Florescu and Cornelia Caragea. 2017. PositionRank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115, Vancouver, Canada. Association for Computational Linguistics.

Ygor Gallina, Florian Boudin, and Beatrice Daille. 2019. KPTimes: A large-scale dataset for keyphrase generation on news documents. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 130–135.

Krishna Garg, Jishnu Ray Chowdhury, and Cornelia Caragea. 2022. Keyphrase generation beyond the boundaries of title and abstract. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5809–5821, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Akash Gautam, Puneet Mathur, Rakesh Gosangi, Debanjan Mahata, Ramit Sawhney, and Rajiv Ratn Shah. 2020. #MeTooMA: Multi-aspect annotations

of tweets related to the MeToo movement. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):209–216.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in COVID-19 tweets. 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 1596–1611.

Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854.

Lara Grimminger and Roman Klinger. 2021. Hate towards the political opponent: A Twitter corpus study of the 2020 US elections on the basis of offensive speech and stance detection. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 171–180.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2021. Cross-domain label-adaptive stance detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9011–9028.

Momchil Hardalov, Arnav Arora, Preslav Nakov, and Isabelle Augenstein. 2022. A survey on stance detection for mis- and disinformation identification. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1259–1277.

Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? Identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 751–762.

Muhammad Naeem Khan, Muhammad Azeem Ashraf, Donald Seinen, Kashif Ullah Khan, and Rizwan Ahmed Laar. 2021. Social media for knowledge acquisition and dissemination: The impact of the COVID-19 pandemic on collaborative learning driven social media adoption. *Frontiers in Psychology*, 12.

Dilek Küçük and Fazli Can. 2020. Stance detection: A survey. *ACM Comput. Surv.*, 53(1):1–37.

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.

Yingjie Li and Cornelia Caragea. 2019. Multi-task stance detection with sentiment and stance lexicons. 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 6298–6304.

Yingjie Li and Cornelia Caragea. 2021a. A multi-task learning framework for multi-target stance detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2320–2326.

Yingjie Li and Cornelia Caragea. 2021b. Target-aware data augmentation for stance detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1850–1860.

Yingjie Li, Tiberiu Sosea, Aditya Sawant, Ajith Jayaraman Nair, Diana Inkpen, and Cornelia Caragea. 2021a. P-Stance: A large dataset for stance detection in political domain. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2355–2365.

Yingjie Li, Chenye Zhao, and Cornelia Caragea. 2021b. Improving stance detection with multi-dataset learning and knowledge distillation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6332–6345.

Yingjie Li, Chenye Zhao, and Cornelia Caragea. 2023. Tts: A target-based teacher-student framework for zero-shot stance detection. In *Proceedings of the ACM Web Conference 2023*, page 1500–1509.

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*, page 3453–3464.

Bin Liang, Qinglin Zhu, Xiang Li, Min Yang, Lin Gui, Yulan He, and Ruifeng Xu. 2022. 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.

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*.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 582–592, Vancouver, Canada. Association for Computational Linguistics.

Lin Miao, Mark Last, and Marina Litvak. 2020. Twitter data augmentation for monitoring public opinion on COVID-19 intervention measures. In *Proceedings of the 1st Workshop on NLP for COVID-19 (Part 2) at EMNLP 2020*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016a. A dataset for detecting stance in tweets. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3945–3952.

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

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

Krutarth Patel and Cornelia Caragea. 2019. Exploring word embeddings in crf-based keyphrase extraction from research papers. In *Proceedings of the 10th International Conference on Knowledge Capture*, K-CAP '19, page 37–44, New York, NY, USA. Association for Computing Machinery.

Delip Rao and Dean Pomerleau. 2017. Fake news challenge.

Jishnu Ray Chowdhury, Cornelia Caragea, and Doina Caragea. 2019. Keyphrase extraction from disaster-related tweets. In *The World Wide Web Conference*, WWW '19, page 1555–1566, New York, NY, USA. Association for Computing Machinery.

Jishnu Ray Chowdhury, Seo Yeon Park, Tuhin Kundu, and Cornelia Caragea. 2022. KPDROP: Improving absent keyphrase generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4853–4870, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Stance detection benchmark: How robust is your stance detection? *KI - Künstliche Intelligenz*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

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.

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.

Christian Stab, Tristan Miller, Benjamin Schiller, Pranav Rai, and Iryna Gurevych. 2018. Cross-topic argument mining from heterogeneous sources. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3664–3674.

Lucas Sterckx, Cornelia Caragea, Thomas Demeester, and Chris Develder. 2016. Supervised keyphrase extraction as positive unlabeled learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1924–1929, Austin, Texas. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2020. X-Stance: A multilingual multi-target dataset for stance detection. In *Proceedings of the 5th Swiss Text Analytics Conference (SwissText) & 16th Conference on Natural Language Processing (KONVENS)*.

Wikipedia. Wikipedia:list of controversial issues. [Online; accessed 10-December-2012].

Lee Xiong, Chuan Hu, Chenyan Xiong, Daniel Campos, and Arnold Overwijk. 2019. Open domain web keyphrase extraction beyond language modeling. 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 5174–5183.

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.

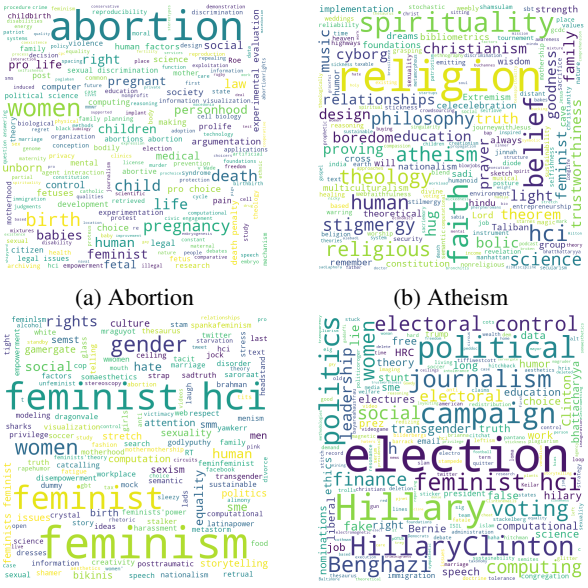
Jiacheng Ye, Tao Gui, Yichao Luo, Yige Xu, and Qi Zhang. 2021. One2Set: Generating diverse keyphrases as a set. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4598–4608.

Xingdi Yuan, Tong Wang, Rui Meng, Khushboo Thaker, Peter Brusilovsky, Daqing He, and Adam Trischler. 2020. One size does not fit all: Generating and evaluating variable number of keyphrases. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7961–7975.

Bowen Zhang, Min Yang, Xutao Li, Yunming Ye, Xiaofei Xu, and Kuai Dai. 2020. Enhancing cross-target stance detection with transferable semantic-emotion knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3188–3197.

A Curation of *Unrelated* target samples

We retrieved a collection of tweets using Twitter API for some controversial topics such as *Black*



(a) Abortion (b) Atheism

(c) Feminist Movement (d) Hillary Clinton

Figure 3: Wordclouds of generated keyphrases for different targets of SemEval-2016 dataset.

Lives Matter, *Communism*, *Conservatism*, *Morality*, etc. The controversial topics were collected from Wikipedia. We manually removed the topics that are related to the targets of our *merged* and *zero-shot* datasets. Further, we performed the following preprocessing steps: (1) We removed the duplicates and retweets. (2) We removed the topics that appear in less than 100 tweets. (3) We removed the tweets that contain any explicit mentions of the targets of our merged and zero-shot datasets. (4) We created the train, validation and test sets following an 80/10/10 split for each topic. Thus, we curated a filtered collection for *Unrelated* samples. Note that *Unrelated* samples used in the merged and zero-shot datasets are not overlapped and examples of *Unrelated* category are shown in Table 9.

Topic	Tweet
Black Lives Matter	Black Lives Matter Proclaims Thanksgiving Is A Holiday Of Colonization On Stolen Land
Communism	We are told that communism causes famines. But it is actually capitalism, colonialism & imperialism that cause food insecurity and mass hunger.
Conservatism	Conservatism isn't about freedoms it's all about control.
Morality	To place morality above compassion or law before love is to nullify nature and scorn nurture. Love knows no wrong.

Table 9: Examples from *Unrelated* category samples.

B Generated Keyphrases in Target Generation Task

As discussed in §6.1, target generation models produce worse performance than target classification models in target identification task. The reason could be that the generated keyphrases might be related to other topics contained in the sentence, which are not correctly mapped to the golden targets in target identification task.

In Figure 3, we show the wordclouds for the generated keyphrases using our keyphrase generation models as described in §4.1 and §6.1. For instance, for the ground truth label *Atheism*, the generated keyphrases are *spirituality*, *religion*, *faith*, *belief*, *philosophy*, etc. We can observe that these generated keyphrases are semantically related to the ground truth target *Atheism* and these generated keyphrases could further be used for other research purposes such as data augmentation of stance detection and multi-target stance annotation.

ACL 2023 Responsible NLP Checklist

A For every submission:

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

☑ A2. Did you discuss any potential risks of your work?
9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

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

B ☑ Did you use or create scientific artifacts?

3.2

☑ B1. Did you cite the creators of artifacts you used?
3.2

☐ 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.
3.2

C ☑ Did you run computational experiments?

8

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
8

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?
we use the default parameters without hyperparameter tuning

☑ 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?
6

☑ 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.)?
Appendix

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.?
Not applicable. Left blank.

☐ 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)?
Not applicable. Left blank.

☐ 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?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.