MLSD: A Few-Shot Learning Framework for Domain Adaptation in Stance Detection

Parush Gera and Tempestt Neal

Bellini College of Artificial Intelligence, Cybersecurity and Computing, University of South Florida, Tampa, Florida, USA
{parush,tjneal}@usf.edu

This paper introduces **M**etric **L**earning-Based Few-Shot Learning for Cross-Target and Cross-Domain **S**tance **D**etection (**MLSD**), a novel framework for adapting stance detection models to new targets and domains with limited labeled data. Stance detection is a text classification task that identifies the position (e.g., favor, against, neutral) an author expresses toward a specific target. However, adapting models to new targets or domains (domain adaptation) is challenging when annotated data is scarce. MLSD addresses this few-shot adaptation problem by using metric learning with triplet loss to identify semantically representative examples and construct a discriminative embedding space for transfer. We evaluate MLSD on two benchmark datasets and six deep learning models widely used in stance detection research, demonstrating statistically significant macro-F1 improvements over standard and random few-shot baselines.

1 Introduction

Stance detection—the task of identifying an author's position toward a given target, such as a person, entity, or issue—has gained significant attention, particularly following its inclusion in Task 6: Stance Detection on Social Media Posts at the 2016 International Workshop on Semantic Evaluation (SemEval 2016) [14]. Common stance categories include *in favor of*, *against*, and *neutral/neither*, though more recent studies have expanded this taxonomy to include labels such as *support*, *agree/disagree*, *refute*, *discussion*, *commenting*, and *unrelated* [3]. For instance, in the SemEval 2016 dataset, the statement "*Whether someone wants to have children or not should be completely up to the person carrying that pregnancy*" conveys a *favor* stance toward the target 'Legalization of Abortion.'

Traditional stance detection assumes training and testing on the same target, but real-world tasks often require generalization across targets. This has led to two subproblems: cross-target stance detection (CTSD) and cross-domain stance detection (CDSD) [2, 20, 18, 3]. CTSD trains on one target and predicts stance on a related one (e.g., training on 'Apple iPhones' and testing on 'Apple MacBooks'). CDSD extends this by crossing domains. For example, previous studies has grouped SemEval 2016 dataset targets of 'Feminist Movement' and 'Legalization of Abortion' under the domain of 'Women's Rights'; a CDSD task might train on 'Women's Rights' and test on the unrelated domain of 'Entertainment' [20].

Although CTSD and CDSD have opened promising avenues for extending stance detection to more realistic scenarios, they also introduce substantial challenges that hinder effective generalization across targets and domains. The wide variation in targets and domains makes training separate models impractical and labeling new data costly [5, 3]. Although some methods use external knowledge to improve generalization [12, 7, 19], they may struggle to capture domain-specific patterns—an issue seen in models like SiamNet and bi-conditional encoders [15, 2, 3], which tend to overfit to source data and transfer poorly.

Transfer learning is a promising approach for improving model transferability in both CTSD and CDSD by leveraging knowledge from a source target or domain to enhance performance on a new one. Unlike training from scratch, which requires large labeled datasets and significant resources, transfer learning enables efficient adaptation through previously learned representations. However, real-world applications often have limited labeled data for new targets or domains. To address this, transfer learning can be combined with few-shot adaptation, where models are fine-tuned using a small but representative sample set. The effectiveness of this strategy hinges on selecting the most informative and context-relevant examples. Current few-shot techniques in stance detection typically rely on random or heuristic-based selection [8, 10, 7], which limits generalization. In contrast, we propose the framework **MLSD** (**M**etric **L**earning-Based Few-Shot Learning for Cross-Target and Cross-Domain **S**tance **D**etection), which integrates metric learning-based sample selection with transfer learning to improve adaptability and performance in both CTSD and CDSD tasks.

To implement this approach, MLSD uses metric learning to assess similarity between text samples from different targets, enabling the selection of a small, representative set of destination samples ("few shots") for fine-tuning. This strategy constructs a task-specific embedding space that is sensitive to both domain and target variation—placing similar examples closer and dissimilar ones farther apart [17]. The selected few-shot samples are then used to adapt a stance model initially trained on a source target, improving its ability to generalize across domains and targets. Importantly, because MLSD is a model-agnostic few-shot sample selection framework that can be integrated with any existing CTSD or CDSD architecture, we demonstrate its effectiveness on two datasets, covering four targets and two domains, using six deep learning models explored in prior literature for stance detection. [1]

2 Background

Stance detection faces two key adaptation challenges: cross-target and cross-domain scenarios. Earlier work has proposed conditional encoders [2], domain-aligned transfer [20], shared-topic matching [18], and target-adaptive graphs [11]. However, these approaches often rely on stable domain overlap, explicit topic labels, or costly graph construction. Models like SiamNet and BiCond [3] can

[1] MLSD implementation is available at https://github.com/parushgera/mlsd-few-shot.

generalize poorly, and adversarial methods with label embeddings [6] or topic-guided contrastive learning require additional computation and accurate topic annotations.

Few-shot learning helps adapt stance models to new targets or domains using only a small labeled sample. Yet, many few-shot methods trade off efficiency or generalizability: commonsense features [12], LLM prompt-tuning [7], conditional generation [19], and semi-supervised variants often introduce extra overhead and rely on randomly chosen shots.

Metric learning offers a promising solution for few-shot settings by constructing representation spaces that capture semantic similarities between objects [17]. By learning to measure similarity, metric learning enables the selection of representative examples, which supports more effective few-shot adaptation. Building on this, MLSD combines metric learning with similarity-based sample selection to improve CTSD and CDSD performance without relying on external resources.

3 Overview of MLSD

3.1 Step 1: Triplet Selection

MLSD uses metric learning to identify semantically representative samples from the destination dataset for few-shot adaptation. Given a large destination dataset of size Z, a small subset $n \ll Z$ is selected based on semantic similarity to the source target. To achieve this, MLSD constructs a task-specific embedding space using a triplet-based objective, which encourages embeddings of similar examples to be closer together and dissimilar examples to be pushed apart.

Each triplet consists of three elements: a source target sample (anchor), another source sample (positive), and an unrelated target sample (negative) that is contextually unrelated. The model learns to distinguish related from unrelated samples by minimizing the triplet loss. To enhance the embedding's discriminative capacity, MLSD incorporates hard negative mining; instead of selecting negatives randomly, top-k challenging candidates are retrieved using a pre-trained encoder and one is selected as the hard negative. This focuses training on the most informative contrasts and improves generalization. To compute these distances, each element in a triplet—source (S), positive (P), or negative (N)—is encoded into an embedding vector, is encoded into an embedding vector, $e(x)$, using a contextual encoder such as BERT. The triplet loss is defined as:

$$L_{\text{triplet}} = \max\big(0,\, d(e(A), e(P)) - d(e(A), e(N)) + m\big),$$

where d is the Euclidean distance and m is a margin hyperparameter enforcing separation between positive and negative pairs.

In practical terms, this step essentially trains a *similarity detection model* that learns whether any text sample is similar to examples in the source dataset. This learned similarity is critical for the next stages as it enables the selection of a small number of destination samples that closely match the source data, without requiring training a full stance detection model on destination data that might be either insufficient or prohibitively large to label.

Notably, this step is model-agnostic and can be implemented using any suitable pre-trained encoder and standard optimizers such as Adam. In our experimental setup, BERT-based embeddings, SBERT for hard negative mining, a margin of 1.0, a learning rate of 5×10^{-5}, a batch size of 64, and early stopping over 10 epochs were used; these parameters can be adapted as needed for different datasets and applications. We note that the quality of the embedding space depends on the encoder; alternative or domain-adapted encoders may be substituted to suit specific applications.

3.2 Step 2: Top-N Few-Shot Sample Selection

The trained similarity detection model is then used to rank destination samples and select the most informative few-shot examples for fine-tuning. For each stance class c, destination samples D_c are scored by how closely they align with the learned source representation:

$$\text{Top-}N(c) = \arg\max_{i=1}^{N} f_\theta(d_i), \quad d_i \in D_c,$$

where $f_\theta(d_i)$ denotes the similarity detection model's confidence score for sample d_i under learned parameters θ. In practice, this means the model from Step 1 filters and ranks new target samples to find those "closest" to the source, ensuring that only the most relevant few are used for transfer learning. By prioritizing representative examples, MLSD reduces the amount of manual labeling needed while retaining high-quality samples for adaptation.

This stage assumes a small amount of labeled destination data per stance class, consistent with the few-shot setting; fully unsupervised domain transfer is out of scope. If classes are highly imbalanced, N can be adjusted per class or all available labeled examples can be used for under-resourced classes.

For large destination datasets, scalability can be improved via approximate nearest neighbor indexing or candidate pre-filtering. As with any similarity-based approach, performance depends on sufficient semantic overlap between source and destination. In cases of extreme divergence, performance may decline, although experiments show consistent improvements even across different domains.

3.3 Step 3: Cross-Target/Cross-Domain Stance Detection

In the final stage, a stance classifier—originally trained on the source target—is fine-tuned with the selected few-shot samples from the destination target's training set, then evaluated on its test set. These semantically aligned examples act as high-quality proxies for the destination domain, helping the classifier adapt to new targets or domains with minimal additional labeled data. By leveraging similarity detection and targeted few-shot selection, this step upgrades the stance model just enough to generalize well without requiring large new datasets.

4 Experimental Methodology

We evaluated the MLSD framework for both CTSD and CDSD using two datasets—the SemEval 2016 Task 6 Stance Detection dataset [14] and the Will-They-Won't-They (WT-WT) dataset [3], the largest publicly available English-language dataset for stance detection consisting of tweets discussing mergers and acquisitions in the healthcare and entertainment industries (Tables 1 and 2).

Table 1: Data distribution in the SemEval-2016 dataset [14].

Target	**Training Data**				**Test Data**			
	%Favor	%Against	%Neutral	#Train	%Favor	%Against	%Neutral	#Test
Atheism (AT)	17.93	59.26	22.81	513	14.55	72.73	12.73	220
Climate Change (CC)	53.67	3.80	42.53	395	72.78	6.51	20.71	169
Feminist Movement (FM)	31.63	49.40	18.98	664	20.35	64.21	15.44	285
Hillary Clinton (HC)	17.13	57.04	25.83	689	15.25	58.31	26.44	295
Legalization of Abortion (LA)	18.53	54.36	27.11	653	16.43	67.50	16.07	280
Donald Trump (DT)	20.94	42.26	36.80	530	20.90	42.37	36.73	177
Total	**25.10**	**47.03**	**27.87**	**3,444**	**23.94**	**55.36**	**20.70**	**1,427**

Table 2: Data distribution in the Will They Won't They dataset [3].

Target	**Training Data**					**Test Data**				
	%Support	%Refute	%Comment	%Unrelated	Total	%Support	%Refute	%Comment	%Unrelated	Total
Healthcare (HLT)	16.10	11.71	37.76	34.33	22,101	16.11	11.71	37.76	34.33	7,367
Entertainment (ENT)	47.74	2.45	8.23	41.75	11,141	47.74	2.45	8.23	41.75	3,714
Total	**13.48**	**8.55**	**41.10**	**36.85**	**33,242**	**13.48**	**8.55**	**41.10**	**36.84**	**11,081**

MLSD was compared against two baselines: (1) a standard setup that trains on a source target and tests directly on a destination target without any few-shot adaptation, and (2) a few-shot approach that selects adaptation samples randomly. These *Standard Training* and *Random Selection* baselines reflect common practice in prior few-shot learning studies [8] and provide a fair basis for isolating the effect of MLSD's similarity-based selection.

The evaluation covers six deep learning models used in prior stance detection work—four RNN-based, one CNN-based, and one BERT-based model [20, 13, 9, 4]—all used in their original, unmodified form. By keeping model architectures and hyperparameters fixed, we ensure that any performance differences arise solely from the sample selection strategy. This controlled design allows a clear comparison between MLSD and standard or random approaches under consistent conditions. These include:

- **BiLSTM** [16]: Encodes text bidirectionally using LSTMs to capture contextual information from both directions.
- **BiCond** [2]: Uses a BiLSTM to encode targets and passes final states to another BiLSTM for text encoding.
- **CrossNet** [20]: A BiLSTM model for CTSD with aspect attention to highlight target-relevant text.

- **TAN** [4]: Combines a bidirectional RNN and BiLSTM with target-specific attention.
- **TextCNN** [9]: A CNN model for sentence classification using pre-trained word embeddings.
- **RoBERTa** [13]: A transformer model pre-trained with masked language modeling, optimized over BERT.

The MLSD-based few-shot selection was evaluated for $n \in \{5, 10, 15\}$, where n is the number of few-shot samples used for domain adaptation. Notably, even when selecting as few as 5, 10, or 15 samples, the proportion of destination data used remains remarkably small: only about 0.02% of FM, LA, and HC, 0.03% of DT, 0.001% of ENT, and just 0.0006% of HLT compared to typical 100–400-shot baselines [8]. Experiments used multiple source targets (FM, LA, HC, and DT), each paired with AT as an unrelated (negative) target during Step 1 for hard negative mining. For CDSD, the ENT and HLT targets were paired with a *Politics* domain constructed by combining HC and DT as the unrelated domain.

For robust and unbiased results, the standard training, random selection, and MLSD pipeline—including triplet-based metric learning, few-shot selection, and stance model fine-tuning—was run independently five times using different random seeds. Although this means our baseline results may appear lower than single best-run numbers reported elsewhere, this approach provides a more reliable estimate of typical model performance and avoids overstating results due to random chance. For each seed, the similarity detection model was trained from scratch, the top-n few-shot samples were selected anew, and the stance detection model was fine-tuned and evaluated on the destination target. Final macro-F_1 scores are reported as the mean across these five fully independent runs, providing a reliable estimate of typical performance than reporting only the best single run, as sometimes done in prior work [8, 1, 21].

5 Results

Tables 3 and 4 present the macro-average F_1-scores for CTSD. In all tables, source → destination indicates the direction of domain or target transfer. Across all models, integrating the MLSD framework consistently outperformed both the standard approach and random sample selection. For cross-domain settings using the WT-WT dataset, the original authors [3] reported poor generalization for the ENT and HLT targets, with F_1-scores of 37.77 for HLT → ENT and 33.62 for ENT → HLT. As shown in Table 5, MLSD led to substantial improvements in these scenarios. Additional CDSD experiments (Table 6) within the SemEval dataset further demonstrate MLSD's effectiveness when the source and destination targets are contextually distinct. On average, MLSD increased performance by 11.72% for FM → HC, 7.20% for FM → DT, 31.32% for ENT → HLT, 29.87% for HLT → ENT, 7.27% for LA → FM, 6.56% for FM → LA, 6.47% for HC → DT, and 7.33% for DT → HC, with values averaged across all classifiers. These results show that MLSD is highly effective in enhancing stance

detection performance, particularly in cross-domain scenarios where traditional methods often struggle.

An interesting observation is that the RoBERTa model did not achieve gains as large as the CNN- and RNN-based models. This may reflect RoBERTa's reliance on extensive pre-training, which already captures much of the generalization needed for stance detection, leaving less room for improvement from MLSD. In contrast, the CNN and RNN models, which do not benefit from such large-scale pre-training, gain more from MLSD's targeted few-shot selection, enhancing their ability to generalize across both cross-target and cross-domain settings.

Table 3: CTSD macro-average F_1-score across five seeds and $n \in 5, 10, 15$ shots. MLSD significantly outperforms random selection for LA $\rightarrow$ FM and FM $\rightarrow$ LA ($p < 0.05$).

	LA → FM			**FM → LA**		
Model	**Standard**	**Random**	**MLSD**	**Standard**	**Random**	**MLSD**
BiCond	36.63%	31.84%	**43.30%**	35.13%	30.94%	**42.67%**
BiLSTM	35.99%	33.03%	**42.20%**	38.73%	29.69%	**39.98%**
CrossNet	35.18%	37.01%	**41.61%**	38.69%	37.80%	**39.72%**
RoBERTa	28.79%	30.56%	**32.15%**	27.88%	28.47%	**30.95%**
TAN	35.68%	35.77%	**39.96%**	36.00%	35.54%	**38.89%**
TextCNN	26.69%	30.07%	**42.38%**	26.16%	32.50%	**42.11%**

Table 4: CTSD macro-average F_1-score across five seeds and $n \in \{5, 10, 15\}$ shots. MLSD significantly outperforms random selection for both HC $\rightarrow$ DT and DT $\rightarrow$ HC ($p < 0.05$, paired t-test).

	HC → DT			**DT → HC**		
Model	**Standard**	**Random**	**MLSD**	**Standard**	**Random**	**MLSD**
BiCond	36.35%	31.78%	**42.59%**	36.70%	30.44%	**43.04%**
BiLSTM	38.53%	30.77%	**40.64%**	38.15%	31.17%	**41.46%**
CrossNet	35.09%	37.39%	**40.47%**	34.98%	36.17%	**40.84%**
RoBERTa	29.21%	30.07%	**31.58%**	29.53%	30.28%	**31.71%**
TAN	36.41%	35.14%	**37.79%**	35.01%	35.19%	**38.91%**
TextCNN	26.79%	30.77%	**41.82%**	26.25%	30.41%	**41.73%**

Table 5: CDSD macro-average F_1-score across five seeds and $n \in \{5, 10, 15\}$ shots. MLSD significantly outperforms random selection for both ENT → HLT and HLT → ENT ($p < 0.05$, paired t-test).

	ENT → HLT			**HLT → ENT**		
Model	**Standard**	**Random**	**MLSD**	**Standard**	**Random**	**MLSD**
BiCond	29.84%	22.36%	**71.92%**	36.46%	28.55%	**68.92%**
BiLSTM	31.43%	34.19%	**71.51%**	35.80%	30.95%	**68.49%**
CrossNet	36.13%	39.30%	**70.79%**	33.71%	33.29%	**67.89%**
RoBERTa	21.51%	24.26%	**24.94%**	20.67%	23.23%	**24.06%**
TAN	28.66%	32.87%	**62.82%**	34.63%	31.10%	**62.91%**
TextCNN	29.48%	27.81%	**66.14%**	30.82%	25.64%	**60.31%**

Table 6: CDSD macro-average F_1-score across five seeds and $n \in \{5, 10, 15\}$ shots. MLSD significantly outperforms random selection for both FM → HC and FM → DT ($p < 0.05$, paired t-test).

	FM → HC			**FM → DT**		
Model	**Standard**	**Random**	**MLSD**	**Standard**	**Random**	**MLSD**
BiCond	28.51%	35.67%	**44.84%**	34.00%	28.35%	**40.71%**
BiLSTM	33.45%	27.01%	**47.86%**	34.86%	29.19%	**35.02%**
CrossNet	30.05%	30.57%	**45.60%**	32.08%	29.66%	**38.27%**
RoBERTa	29.20%	32.85%	**34.62%**	31.76%	31.25%	**34.28%**
TAN	33.37%	32.93%	**42.76%**	36.34%	32.23%	**36.75%**
TextCNN	22.25%	28.63%	**42.35%**	23.79%	28.35%	**36.79%**

6 Conclusion and Limitations

This work introduced MLSD, a novel metric learning-based framework for few-shot cross-target and cross-domain stance detection. By combining triplet loss with hard negative mining, MLSD effectively identifies the most informative destination samples to fine-tune pre-trained stance models, consistently outperforming random selection—especially when targets or domains differ significantly. The results show that robust stance detection can be achieved with minimal additional labeling effort, requiring only a handful of carefully selected examples.

However, MLSD has important practical considerations. Its performance depends on the quality of the pre-trained embeddings, and the hard negative mining step adds computational overhead during the similarity training phase. Although MLSD greatly reduces annotation needs, it still requires access to a small number of labeled examples for each destination target to enable effective few-shot adaptation. Future work could explore fully unsupervised extensions, alter-

native encoder architectures, or integration with large language models to further improve generalization while keeping the framework lightweight. Overall, MLSD demonstrates that metric learning can meaningfully advance few-shot stance detection in realistic cross-target and cross-domain scenarios where labeled data is limited.

References

1. Arakelyan, E., Arora, A., Augenstein, I.: Topic-guided sampling for data-efficient multi-domain stance detection. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 13448–13464. Association for Computational Linguistics, Toronto, Canada (Jul 2023), https://aclanthology.org/2023.acl-long.752
2. Augenstein, I., Rocktäschel, T., Vlachos, A., Bontcheva, K.: Stance detection with bidirectional conditional encoding. In: Su, J., Duh, K., Carreras, X. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 876–885. Association for Computational Linguistics, Austin, Texas (Nov 2016), https://aclanthology.org/D16-1084/
3. Conforti, C., Berndt, J., Pilehvar, M.T., Giannitsarou, C., Toxvaerd, F., Collier, N.: Will-they-won't-they: A very large dataset for stance detection on Twitter. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1715–1724. Association for Computational Linguistics, Online (Jul 2020), https://aclanthology.org/2020.acl-main.157
4. Du, J., Xu, R., He, Y., Gui, L.: Stance classification with target-specific neural attention networks. In: 26th International Joint Conference on Artificial Intelligence, IJCAI 2017. pp. 3988–3994. International Joint Conferences on Artificial Intelligence (2017)
5. Gera, P., Neal, T.: A comparative analysis of stance detection approaches and datasets. In: Deutsch, D., Udomcharoenchaikit, C., Opitz, J., Gao, Y., Fomicheva, M., Eger, S. (eds.) Proceedings of the 3rd Workshop on Evaluation and Comparison of NLP Systems. pp. 58–69. Association for Computational Linguistics, Online (Nov 2022), https://aclanthology.org/2022.eval4nlp-1.7
6. Hardalov, M., Arora, A., Nakov, P., Augenstein, I.: Cross-domain label-adaptive stance detection. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 9011–9028. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021), https://aclanthology.org/2021.emnlp-main.710
7. Jiang, Y., Gao, J., Shen, H., Cheng, X.: 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. p. 837–847. SIGIR '22, Association for Computing Machinery, New York, NY, USA (2022), https://doi.org/10.1145/3477495.3531979
8. Khiabani, P.J., Zubiaga, A.: Few-shot learning for cross-target stance detection by aggregating multimodal embeddings. IEEE Transactions on Computational Social Systems pp. 1–10 (2023)

9. Kim, Y.: Convolutional neural networks for sentence classification. In: Moschitti, A., Pang, B., Daelemans, W. (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1746–1751. Association for Computational Linguistics, Doha, Qatar (Oct 2014), https://aclanthology.org/D14-1181
10. Li, Y., Zhao, C., Caragea, C.: Tts: A target-based teacher-student framework for zero-shot stance detection. In: Proceedings of the ACM Web Conference 2023. p. 1500–1509. WWW '23, Association for Computing Machinery, New York, NY, USA (2023), https://doi.org/10.1145/3543507.3583250
11. Liang, B., Fu, Y., Gui, L., Yang, M., Du, J., He, Y., Xu, R.: Target-adaptive graph for cross-target stance detection. In: Proceedings of the Web Conference 2021. p. 3453–3464. WWW '21, Association for Computing Machinery, New York, NY, USA (2021), https://doi.org/10.1145/3442381.3449790
12. Liu, R., Lin, Z., Tan, Y., Wang, W.: Enhancing zero-shot and few-shot stance detection with commonsense knowledge graph. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. pp. 3152–3157 (2021)
13. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692 (2019), http://arxiv.org/abs/1907.11692
14. Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., Cherry, C.: Semeval-2016 task 6: Detecting stance in tweets. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). pp. 31–41 (2016)
15. Santosh, T.Y., Bansal, S., Saha, A.: Can siamese networks help in stance detection? In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data. p. 306–309. CODS-COMAD '19, Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3297001.3297047
16. Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45(11), 2673–2681 (1997)
17. Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M.: Generalizing from a few examples: A survey on few-shot learning. ACM Comput. Surv. 53(3) (Jun 2020), https://doi.org/10.1145/3386252
18. Wei, P., Mao, W.: Modeling transferable topics for cross-target stance detection. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1173–1176. SIGIR'19, Association for Computing Machinery, New York, NY, USA (2019), https://doi.org/10.1145/3331184.3331367
19. Wen, H., Hauptmann, A.: Zero-shot and few-shot stance detection on varied topics via conditional generation. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 1491–1499. Association for Computational Linguistics, Toronto, Canada (Jul 2023), https://aclanthology.org/2023.acl-short.127
20. Xu, C., Paris, C., Nepal, S., Sparks, R.: Cross-target stance classification with self-attention networks. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 778–783. Association for Computational Linguistics, Melbourne, Australia (Jul 2018), https://aclanthology.org/P18-2123
21. Zhou, Y., Cristea, A.I., Shi, L.: Connecting targets to tweets: Semantic attention-based model for target-specific stance detection. In: Web Information Systems Engineering–WISE 2017: 18th International Conference, Puschino, Russia, October 7-11, 2017, Proceedings, Part I 18. pp. 18–32. Springer (2017)