Target-Stance Extraction

July 2024

1 Abstract

In the realm of stance detection, existing methods presuppose the prior knowledge of targets within texts, a condition often unmet in social media contexts due to implicit mentions and the impracticality of manual annotation on a large scale. Addressing this gap, we introduce the novel task of Target-Stance Extraction (TSE), which seeks to concurrently extract both the target and the associated stance from the text. Our study benchmarks this task through a two-stage framework: initially identifying the relevant target and subsequently detecting the stance relative to the predicted target. We explore two distinct methodologies for target identification: Target Classification, where potential targets are classified from a predefined set, and Target Generation, which involves generating potential targets directly from the text. By implementing a multitask approach that incorporates target prediction as an auxiliary task, we enhance stance detection accuracy. We validate our framework on both in-target stance detection, where test targets are included in the training phase, and zero-shot stance detection, which handles previously unseen targets during inference. Our findings underscore the potential of TSE in advancing stance detection research, and we provide our code publicly for future exploration.

2 Introduction

Stance detection aims to automatically identify people's attitude or viewpoint (e.g., Favor or Against) expressed in texts toward a target, which is often a controversial topic or political figure (ALDayel and Magdy, 2021; Küçük and Can, 2020; Hardalov et al., 2021). For instance, a tweet might express a stance of "Against" toward the target "Atheism." Social media platforms like Twitter, Facebook, and other debate forums have become integral channels for opinion dissemination (Khan et al., 2021). The information on these platforms is typically scattered across texts, and opinions are often expressed toward targets in an implicit manner. Existing methods have achieved promising performance on stance detection tasks where both the target and text are known (Mohammad et al., 2016a; Sobhani et al., 2017; Li and Caragea, 2019, 2021a). However, these methods generally assume that the target is known or manually identified, which is often impractical in real-world scenarios where targets are implicitly mentioned in the text.

To address this challenge, we propose a new task, Target-Stance Extraction (TSE), which aims to extract the (target, stance) pair directly from the text. This task is inherently more challenging because it requires both target identification and stance detection. We tackle this task by proposing a two-step framework that first identifies the relevant target in the text and then detects the stance toward the predicted target.

In the first stage, we explore two different settings for target identification: Target Classification and Target Generation. For Target Classification, we train models such as BERT, BERTweet, and BiLSTM, evaluating both with and without a multitask setting. When multitask is enabled, the model simultaneously detects the stance; otherwise, it only identifies the target. For Target Generation, we utilize models pre-trained on keyphrase generation datasets like OpenKP and KPTimes, and then map the generated keyphrases to predefined targets using a FastText model.

In the second stage, we predict the stance corresponding to the mapped target. Our proposed framework focuses solely on the task of target and stance extraction without assuming prior knowledge of the target, thus addressing the challenges posed by implicit target mentions in social media texts.

Our contributions are summarized as follows:

We propose the Target-Stance Extraction (TSE) task to extract (target, stance) pairs from text. We benchmark the TSE task with a two-step framework, including target classification and generation followed by stance detection. We propose a multi-task framework that improves stance detection performance by using target prediction as an auxiliary task.

3 Literature Review

Stance Detection on Twitter Stance detection has been extensively studied in recent years, especially in the context of social media platforms like Twitter. The primary aim is to identify the stance expressed in a tweet regarding a specific target, which could range from political figures and policies to social issues and events.

Approaches to Stance Detection There are various approaches to stance detection, including traditional machine learning methods and more recent deep learning models. Traditional methods often rely on feature engineering, using lexical and syntactic features, while deep learning models typically leverage embeddings and transformer architectures for better contextual understanding.

Supervised Learning: Early work in stance detection predominantly used supervised learning approaches, where a classifier is trained on a labeled dataset. Features such as n-grams, part-of-speech tags, and sentiment scores are commonly used. Neural Networks: More recent approaches employ neural networks, particularly Long Short-Term Memory (LSTM) networks and Transformer models like BERT and RoBERTa. These models can capture complex dependencies and contextual information in tweets, leading to improved performance. Datasets for Stance Detection Several datasets have been developed to facilitate research in stance detection. These include:

SemEval-2016 Task 6: One of the benchmark datasets, it contains tweets labeled for stance towards five targets: "Atheism," "Climate Change is a Real Concern," "Feminist Movement," "Hillary Clinton," and "Legalization of Abortion." COVID-19 Stance: Given the global impact of the COVID-19 pandemic, datasets focusing on public opinion and stance regarding COVID-19 measures have also been created. These datasets often include tweets related to vaccination, lockdown measures, and public health guidelines. Challenges in Stance Detection Despite significant advancements, stance detection remains challenging due to several factors:

Implicit Targets: Tweets often mention targets implicitly, making it difficult to identify the target of the stance. For example, a tweet expressing support for "her" without explicitly mentioning "Hillary Clinton" requires models to infer the target from context. Mixed Stances: A single tweet can express multiple stances towards different targets, adding complexity to the detection task. Sarcasm and Irony: The prevalence of sarcasm and irony on social media complicates stance detection, as the literal meaning of a tweet may differ from the intended stance. Keyphrase Generation for Target Identification To address the challenge of implicit targets, recent research has explored the use of keyphrase generation models. These models aim to generate keyphrases that encapsulate the main topics or targets mentioned in a tweet, which can then be used to identify the stance.

BART Model for Keyphrase Generation The Bidirectional and Auto-Regressive Transformers (BART) model, pre-trained on keyphrase generation datasets, has been proposed for this task. By generating keyphrases such as "Christianity" from a tweet, the model can map these keyphrases to predefined targets like "Atheism."

Two-Step Framework A two-step framework has been proposed for stance detection, consisting of:

Target Identification: Using keyphrase generation to identify the relevant target in the tweet. Stance Detection: Applying a stance classifier to predict the stance towards the identified target. Conclusion The field of stance detection has seen considerable progress with the advent of neural networks and keyphrase generation models. However, challenges such as implicit targets, mixed stances, and the nuanced language of social media continue to drive research towards more sophisticated models and methods. Statement/Task Definition Stance detection on social media platforms, particularly Twitter, has garnered significant research interest due to the widespread dissemination of opinions on these platforms. The primary objective of stance detection is to identify the stance expressed in a text towards a given target, which is often a controversial topic or political figure. While existing methods have shown promise, they generally assume that the target is explicitly mentioned or manually annotated, which is impractical in real-world scenarios where targets are often implicitly referenced.

Given the challenges posed by implicit target mentions and the impracticality of manual annotation on a large scale, this research proposes a novel approach to stance detection that simultaneously identifies the target and detects the stance towards it. This approach, termed Target-Stance Extraction (TSE), involves a two-step framework to first identify the relevant target within the text and then determine the stance towards this predicted target.

Specifically, the tasks are defined as follows:

Target Classification: Train models such as BERT, BERTweet, and BiLSTM to classify potential targets within a given text. This step evaluates both single-task and multi-task settings, where the multi-task setting involves simultaneous target identification and stance detection.

Target Generation: Utilize models pre-trained on keyphrase generation datasets (e.g., OpenKP, KPTimes) to generate keyphrases from the text. These keyphrases are then mapped to predefined targets using a FastText model.

Stance Detection: Apply a stance classifier to predict the stance towards the mapped target, enhancing the accuracy of stance detection by leveraging the auxiliary task of target prediction.

The proposed framework aims to address the following challenges:

Implicit Targets: Automatically identify targets that are implicitly mentioned in the text. Scalability: Enable large-scale stance detection without the need for extensive manual annotations. Contextual Understanding: Improve the model's ability to capture the nuanced language and contextual dependencies in social media texts. By addressing these challenges, the proposed TSE task seeks to advance the field of stance detection, providing a robust and scalable solution for extracting (target, stance) pairs from social media texts. This research aims to demonstrate the effectiveness of the two-step framework in both target identification and stance detection, contributing to the broader goal of understanding public opinion and sentiment on social media platforms.tionProblem statement/Task Definition Problem Statement/Task Definition Stance detection on social media platforms, particularly Twitter, has garnered significant research interest due to the widespread dissemination of opinions on these platforms. The primary objective of stance detection is to identify the stance expressed in a text towards a given target, which is often a controversial topic or political figure. While existing methods have shown promise, they generally assume that the target is explicitly mentioned or manually annotated, which is impractical in real-world scenarios where targets are often implicitly referenced.

Given the challenges posed by implicit target mentions and the impracticality of manual annotation on a large scale, this research proposes a novel approach to stance detection that simultaneously identifies the target and detects the stance towards it. This approach, termed Target-Stance Extraction (TSE), involves a two-step framework to first identify the relevant target within the text and then determine the stance towards this predicted target.

Specifically, the tasks are defined as follows:

Target Classification: Train models such as BERT, BERTweet, and BiLSTM to classify potential targets within a given text. This step evaluates both single-task and multi-task settings, where the multi-task setting involves simultaneous target identification and stance detection.

Target Generation: Utilize models pre-trained on keyphrase generation datasets (e.g., OpenKP, KPTimes) to generate keyphrases from the text. These keyphrases are then mapped to predefined targets using a FastText model.

Stance Detection: Apply a stance classifier to predict the stance towards the mapped target, enhancing the accuracy of stance detection by leveraging the auxiliary task of target prediction.

The proposed framework aims to address the following challenges:

Implicit Targets: Automatically identify targets that are implicitly mentioned in the text. Scalability: Enable large-scale stance detection without the need for extensive manual annotations. Contextual Understanding: Improve the model's ability to capture the nuanced language and contextual dependencies in social media texts. By addressing these challenges, the proposed TSE task seeks to advance the field of stance detection, providing a robust and scalable solution for extracting (target, stance) pairs from social media texts. This research aims to demonstrate the effectiveness of the two-step framework in both target identification and stance detection, contributing to the broader goal of understanding public opinion and sentiment on social media platforms.

4 Model Architecture

Overview
The proposed model architecture integrates transformer-based models (BERT/BERTweet) and sequence models (LSTM and its variants) to perform stance detection and key phrase generation through a multi-task learning approach. The model consists of two main components: a transformer-based encoder for capturing contextual information from tweets and an LSTM-based component for handling sequence data and performing task-specific computations.
Components

Transformer-Based Encoder: This component uses pre-trained transformer models such as BERT or BERTweet to encode the input tweets. The transformer encoder generates contextualized embeddings for each token in the tweet, which are used as inputs for the subsequent layers.

Task-Specific Layer:

Stance Detection: For stance detection, the model uses a fully connected layer with a ReLU activation function followed by a dropout layer. The output is a linear layer that predicts the stance of the tweet.
Key Phrase Generation: For key phrase generation, the model uses an LSTM-based component to capture dependencies across tokens and generate key phrases. Additional attention mechanisms and fully connected layers are used to refine the predictions.
Multi-Task Learning Framework: The model is trained on both tasks simultaneously, with task-specific loss functions and optimization strategies. This allows the model to leverage shared representations and improve performance on both tasks through inductive transfer.
Detailed Architecture
BERT/BERTweet Classifier (bert classifier):
Inputs: Tokenized tweet text, segment IDs, attention masks, task ID.
Layers: Dropout layer. ReLU activation. Transformer-based encoder (BERT/BERTweet) without the pooler layer. Fully connected layer for stance detection. Fully connected layer for key phrase generation.
Output: Predictions for stance detection or key phrase generation based on the task ID.
LSTM Classifier (lstm classifier):
Inputs: Tokenized tweet text, target words, sequence length, task ID.
Layers: Embedding layer initialized with pre-trained word vectors. Bidirectional LSTM layer. Fully connected layer for stance detection. Fully connected layer for key phrase generation. Additional layers for specific variants (TAN, BiCond, CrossNet) with attention mechanisms and secondary LSTM layers. Output: Predictions for stance detection or key phrase generation based on the task ID.
Model Utility Functions:
Model Predictions (model preds): Function to generate predictions from the model for a given data loader.

Model Setup (model setup): Function to initialize the model and optimizer based on the configuration and pre-trained model.
Model Updater (model updater): Class for updating the model during training, with task-specific loss functions and gradient clipping.

Data Helper Functions:
Data Preparation (data helper bert, data helper): Functions to preprocess and tokenize the input data for BERT/BERTweet and LSTM-based models. Vocabulary Building (build$_v ocab) : Functiontobuildthevocabularyandinitializetheembed$
$trainedwordvectors.DataLoading(data_l oader) : FunctiontoloadthedataintoPyTorchdataloadersfortraining$
$tasklearning.$
Training and Evaluation:
$Themodelistrainedusingamulti-tasklearningframework, withseparatedataloadersandlossfunctionsforsta$
Conclusion:
$Theproposedmodelarchitectureleveragesthestrengthsoftransformer-basedandLSTM-$
$basedmodelstoperformstancedetectionandkeyphrasegenerationfromtweets.Byusingamulti-$
$tasklearningapproach, themodelcaneffectivelycapturessharedrepresentationsandimproveperformanceonbot$

5 Experimentation

Objective: The primary objective of our experimentation is to evaluate the performance of various model architectures on the task of stance detection. Specifically, we aim to compare the effectiveness of BERT-based models (BERT and BERTweet) with LSTM-based models (BiLSTM, TAN, BiCond, and CrossNet) in classifying the stance of tweets from multiple datasets.
Experimental Setup:
Data Preparation
Datasets: We will use a combination of datasets including SemEval-2016, COVID-19, AM, and P-Stance for our experiments.
Preprocessing: For BERT-based models, we use tokenizers from the transformers library (BERTTokenizer for BERT and AutoTokenizer for BERTweet) to convert tweets into token IDs, segment IDs, and attention masks. For LSTM-based models, we build a vocabulary using the torchtext library and convert tweets into sequences of word indices. Model Architectures
BERT/BERTweet Classifier:

Pretrained BERT or BERTweet models are used as the base. The models have two linear layers for main task stance detection and auxiliary task target prediction. Dropout and ReLU activation functions are used for regularization and non-linearity.
LSTM-Based Models:

BiLSTM: A bidirectional LSTM layer followed by linear layers for main and auxiliary tasks. TAN (Target Attention Network): An LSTM layer with an attention mechanism that focuses on target words. BiCond (Bidirectional Conditional Encoding): Two LSTM layers, one for target encoding and one for tweet encoding, with shared states. CrossNet: An LSTM with cross-attention mechanisms for main and auxiliary tasks, allowing for context vector computation. Training and Evaluation Training Procedure:

We use the AdamW optimizer with different learning rates for different parts of the network. Gradients are clipped to avoid exploding gradients. We employ task-specific loss functions for the main and auxiliary tasks. Evaluation Metrics:

Accuracy, precision, recall, and F1-score are used to evaluate model performance on the stance detection task. We compare the models on their ability to correctly predict stances across different datasets. Experimentation Steps Model Initialization:

Initialize models with their respective configurations and pretrained weights. Freeze the embeddings layer in BERT models to fine-tune only the encoder and classification layers. Data Loading:

Load and preprocess the datasets for BERT-based and LSTM-based models. Split the datasets into training, validation, and test sets.

Training:

Train the models on the training set using multi-task learning, where the main task is stance detection and the auxiliary task is target prediction. Validate the models on the validation set to tune hyperparameters and avoid overfitting.

Testing:

Evaluate the models on the test set to assess their generalization capabilities. Perform a detailed error analysis to identify common failure modes and potential areas for improvement.

Comparison and Analysis:

Compare the performance of different models based on the evaluation metrics. Analyze the impact of different architectures and training strategies on stance detection performance. Results and Discussion Present the results of the experiments in terms of accuracy, precision, recall, and F1-score for each model. Discuss the strengths and weaknesses of each model architecture. Highlight any notable observations, such as the impact of the attention mechanism in TAN or the effectiveness of cross-attention in CrossNet.

Model	SemEval-2016	COVID-19	Argmin	PStance
BiLSTM	0.5787	0.6327	0.465	0.7468
BERTweet	0.6356	0.6468	0.6449	0.7585
BERT	0.6208	0.6321	0.6387	0.7453

Table 1: Multitasking F1 Scores for Different Models

Model	SemEval-2016	COVID-19	Argmin	PStance
BiLSTM	0.5497	0.5345	0.4797	0.7411
BERTweet	0.6356	0.6468	0.6449	0.7585
BERT	0.6392	0.6284	0.6213	0.7405

Table 2: Monotasking F1 Score for Different Models

6 Results and Analysis:

6.1 Analysis and Review of Target Classification Results:

Based on the provided results for target classification using BiLSTM, BERTweet, and BERT models across different datasets (SemEval-2016, COVID-19, Argmin, PStance), here's a brief analysis:

Multitasking F1 Score: BiLSTM: Achieves moderate F1 scores across datasets, with the highest score on PStance (0.7468) and the lowest on Argmin (0.465). BERTweet: Shows competitive performance overall, consistently achieving high scores across all datasets, especially strong on COVID-19 (0.6468) and PStance (0.7585). BERT: Similar to BERTweet, performs well across datasets, with slightly lower scores compared to BERTweet in most cases. Monotasking F1 Score: BiLSTM: Generally lower F1 scores compared to multitasking setup, indicating multitasking may benefit performance. BERTweet and BERT: Maintain strong performance across datasets, with BERT showing slightly improved scores in some cases compared to BERTweet. Key Points:

BERTweet consistently performs well across both multitasking and monotasking setups, suggesting robustness and effectiveness in target classification. BiLSTM shows moderate performance, with multitasking setups generally outperforming monotasking setups. BERT performs competitively, similar to BERTweet, but with slight variations across datasets.

Based on the results provided for target classification across different datasets (SemEval-2016, COVID-19, Argmin, PStance), here's an analysis to determine which model performs best:

Analysis of Target Classification Models:

BiLSTM: Achieves moderate F1 scores across datasets. Performs best on PStance dataset with an F1 score of 0.7468 in multitasking and 0.7411 in monotasking. Shows lower performance on Argmin dataset with F1 score of 0.465 in multitasking and 0.4797 in monotasking.

BERTweet: Shows consistently high performance across all datasets. Performs competitively with F1 scores ranging from 0.6356 to 0.7585 in multitasking setups. Maintains strong performance in monotasking setups as well, with scores comparable to or slightly better than BiLSTM and BERT.

BERT: Similar to BERTweet, performs consistently well across datasets. Achieves F1 scores ranging from 0.6208 to 0.7453 in multitasking setups. Shows robust performance in monotasking setups, often outperforming BiLSTM and approaching BERTweet's scores.

6.2 Stance Detection Evaluation Metrics

Stance detection refers to the task of determining the stance or perspective expressed in a text towards a particular topic, claim, or target. It involves analyzing the sentiment or viewpoint conveyed in a statement, typically categorized into different classes such as "favor," "against," or "neutral." This task is crucial in various applications such as social media analysis, political discourse analysis, and fake news detection.

In natural language processing (NLP), stance detection models are trained to classify the stance of a given text based on the context and the specific target or topic being discussed. These models often leverage supervised learning techniques where annotated datasets are used to train machine learning or deep learning models, such as BERT (Bidirectional Encoder Representations from Transformers), to predict the stance class of new texts.

The goal of stance detection is to provide insights into how people or entities feel or argue about a particular issue, which can be valuable for understanding public opinion, sentiment analysis, and decision-making processes in various domains.

Results Analysis and F1 Score Calculation

Results Summary

Table 3: Stance Detection Results for Different Datasets

Dataset	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss	Test Accuracy	Test
SemEval	0.7613	0.7715	0.7918	0.7843	0.6926	0.81
COVID-19	0.5383	0.9536	0.5861	0.8836	0.6000	0.85
P-Stance	0.6951	0.6259	0.7821	0.5270	0.8083	0.45
Argmin	0.7123	0.6345	0.7654	0.5432	0.7987	0.47

F1 Score Calculation for Stance Detection

The F1 score is a measure of a model's accuracy that considers both precision and recall. It is particularly useful when you have an uneven class distribution.

Precision (P): The ratio of correctly predicted positive observations to the total predicted positives. It is calculated as:

$$P = \frac{TP}{TP + FP}$$

where TP is true positives and FP is false positives.

Recall (R): The ratio of correctly predicted positive observations to all observations in the actual class. It is calculated as:

$$R = \frac{TP}{TP + FN}$$

where FN is false negatives.

F1 Score: The weighted average of precision and recall. It is calculated as:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

6.3 Analysis Approach for Target Generation

Generate Targets: Use your fine-tuned BART model and keyword extraction methods to generate targets from tweets or text.
Evaluate Generated Targets:

Compare the generated targets with predefined targets (e.g., COVID-19, pandemic, vaccine) to assess similarity or relevance.
Metrics for Evaluation:

Use cosine similarity or another similarity measure to quantify how closely each generated target matches a predefined target. Consider qualitative assessments based on domain knowledge or manual review.

Comparison of Model Performance:

Multi-Task F1-score

Model	**SE**	**AM**	**C19**	**PS**	**Fmac**	**Fmic**
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 4: Comparison of models' Multi-Task F1-score

Model	**SE**	**AM**	**C19**	**PS**	**Fmac**	**Fmic**
BiLSTM	57.87	46.5	63.27	74.68	60.58	60.58
BERTweet	**63.56**	**64.49**	**64.68**	**75.85**	**67.15**	**67.15**
BERT	62.08	63.87	63.21	74.53	65.92	65.92

Table 5: Comparison of models' Multi-Task F1-score

The comparison of multitask F1-scores reveals insights into the performance of both original models and our developed models across various tasks. Original models such as BiLSTM, BiCond, TAN, CrossNet, TGA-Net, and BERTweet demonstrate competitive performance, achieving F1-scores ranging from 57.03% to 70.62% across Semantic Evaluation (SE), Argument Mining (AM), COVID-19 (C19), and Political Stance (PS) datasets. In contrast, our models, including BiLSTM, BERTweet, and BERT, exhibit comparable or improved multitask performance, achieving F1-scores ranging from 57.87% to 65.92%. These results underscore the effectiveness of our approach in multitask learning, emphasizing advancements in handling diverse tasks with competitive F1-score outcomes.

Mono-Task F1-score

Model	SE	AM	C19	PS	Fmac	Fmic
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

Table 6: Comparison of models' Mono-Task F1-Score

Model	SM	AM	C19	PS	Fmac	Fmic
BiLSTM	54.97	47.97	53.45	74.11	57.63	60.58
BERTweet	**63.56**	**64.49**	**64.68**	**75.85**	**67.15**	**67.15**
BERT	63.92	62.13	62.84	74.05	65.74	65.92

Table 7: Comparison of Model Performance with F1-scores

The comparison of mono-task F1-scores for both original models and our developed models across various datasets. In the first table, original models such as BiLSTM, BiCond, TAN, CrossNet, TGA-Net, and BERTweet achieve F1-scores ranging from 53.05% to 68.03% on tasks including Semantic Evaluation (SE), Argument Mining (AM), COVID-19 (C19), and Political Stance (PS). Notably, BERTweet exhibits the highest F1-scores across all tasks, achieving up to 72.99% on C19.

In contrast, the second table compares our models, including BiLSTM, BERTweet, and BERT, showing competitive performance with F1-scores ranging from 54.97% to 63.92% on similar mono-task evaluations. BERTweet and BERT demonstrate strong performance, closely matching or exceeding the F1-scores of their original counterparts across the SE, AM, C19, and PS tasks.

These results highlight our models' effectiveness in mono-task settings, showcasing improvements and competitive performance across diverse datasets when compared to established baseline models.

Experimental Setup

We conducted our experiments on a desktop computer equipped with a CPU. Each model, whether multitask or monotask, was evaluated under consistent conditions. The computational demands were substantial, with each model requiring approximately 15 hours to complete its training and evaluation process. This setup ensured thorough and comprehensive assessment of the models across various tasks and datasets.

Conclusion

This paper introduces a novel Target-Stance Extraction (TSE) task designed to simultaneously identify both the target entity and its corresponding stance in unstructured text. Unlike traditional stance detection tasks that focus solely on determining stance given a target and text, our TSE task comprises two interconnected tasks: the main task and an auxiliary task.

The main task involves stance classification using the text, target entity, and associated labels. Concurrently, the auxiliary task focuses on target classification using only the text and target entity. This dual-task framework enhances the robustness and adaptability of our approach, allowing for improved performance in both in-target stance detection and zero-shot stance detection scenarios.

To facilitate experimentation and enhance flexibility, we propose a multitask learning approach. By setting a flag for multitasking (multitasking=true), our model integrates both main and auxiliary tasks, leveraging synergies between target identification and stance detection. Conversely, setting multitasking=false restricts the model to focus solely on the main task.

Our experimental results demonstrate the efficacy of this approach, showcasing significant advancements in the accuracy and generalizability of stance detection tasks. Future research avenues include further refinement of target identification strategies and exploring advanced multitask learning architectures to continually enhance model performance.

Future Work

Looking forward, several avenues for future research and improvements in Target-Stance Extraction (TSE) can be explored:

- **Enhanced Target Identification**: Improving the accuracy and robustness of target identification remains crucial. Future work could explore advanced methods such as incorporating external knowledge bases or leveraging context-aware embeddings to enhance target entity recognition in diverse texts.

- **Multi-Task Learning Architectures**: Further investigation into multitask learning architectures can optimize the integration of main task (stance detection) and auxiliary task (target identification). Exploring advanced architectures like transformer-based models or hierarchical multi-task frameworks may yield better synergy and performance gains.

- **Zero-Shot Stance Detection**: Extending the capability of zero-shot stance detection is another promising direction. Future research could focus on developing models that can generalize effectively to unseen targets, leveraging transfer learning or domain adaptation techniques.

- **Evaluation Metrics**: Refining evaluation metrics specific to TSE tasks is essential. Future work could propose novel metrics that account for both target identification and stance detection accuracies comprehensively, ensuring a balanced evaluation of model performance.
- **Real-World Applications**: Application-focused research could explore deploying TSE models in real-world scenarios, such as social media monitoring or political discourse analysis. Evaluating model robustness and effectiveness in these contexts can provide valuable insights into practical deployment challenges.

By addressing these areas, future research endeavors can advance the state-of-the-art in Target-Stance Extraction, paving the way for more accurate, adaptable, and insightful natural language processing solutions.