Enhancing Stance Detection through Sequential Weighted Multi-task Learning

Nora Alturayeif (nsalturayeif@iau.edu.sa)
King Fahd University of Petroleum and Minerals
Hamzah Luqman
King Fahd University of Petroleum and Minerals
Moataz Ahmed
King Fahd University of Petroleum and Minerals

Research Article

Keywords: Stance detection, Multi-task learning (MTL), Natural Language Processing (NLP), Sentiment analysis, Social media, Sarcasm detection, Opinion mining

Posted Date: August 4th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3218743/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Social Network Analysis and Mining on December 9th, 2023. See the published version at https://doi.org/10.1007/s13278-023-01169-7.

Enhancing Stance Detection through Sequential Weighted Multi-task Learning

Nora Alturayeif[1,2*], Hamzah Luqman[1,3] and Moataz Ahmed[1,4]

[1]Information and Computer Science Department, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia.
[2]Department of Computer Science, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam, 31441, Saudi Arabia.
[3]SDAIA-KFUPM Joint Research Center for Artificial Intelligence, King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia.
[4]Interdisciplinary Research Center of Intelligent Secure Systems (IRC-ISS), King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia.

*Corresponding author(s). E-mail(s): nsalturayeif@iau.edu.sa;
Contributing authors: hluqman@kfupm.edu.sa;
moataz@kfupm.edu.sa;

Abstract

The exponential growth of user-generated content on social media platforms, online news outlets, and digital communication has necessitated the development of automated tools for analyzing opinions and attitudes expressed in text. Stance detection, a critical task in Natural Language Processing (NLP), aims to identify the underlying perspective or viewpoint of an individual or group towards a specific topic or target. This paper explores the challenges of stance detection, particularly in the context of social media, where brevity, informality, and limited contextual information prevail. While sentiment analysis focuses on explicit sentiment polarity, stance detection classifies the stance or viewpoint of a text towards a target, often of an abstract nature. This study introduces two multi-task learning (MTL) models that integrate sentiment analysis and sarcasm detection tasks to enhance stance detection performance. Four task weighting techniques are proposed and evaluated, and their effectiveness in the MTL models is demonstrated. Extensive evaluations on English and Arabic benchmark datasets highlight the advantages of the proposed models. Among them, the multi-target sequential MTL model stands out with its hierarchical weighting approach, as it achieves state-of-the-art performance. The study underscores the potential of MTL in improving stance detection and provides insights into the interaction between sentiment and stance, while considering the impact of sarcasm.

Keywords: Stance detection, Multi-task learning (MTL), Natural Language Processing (NLP), Sentiment analysis, Social media, Sarcasm detection, Opinion mining

1 Introduction

The vast growth of social media platforms, online news outlets, and digital communication has led to an exponential increase in user-generated content in recent years. This unprecedented surge in online discourse has sparked an urgent need to develop automated tools and techniques capable of effectively analyzing the opinions and attitudes expressed within these expansive streams of text. Stance detection, a critical task within the field of Natural Language Processing (NLP), aims to identify the position or perspective of a writer towards a specific topic or entity by analyzing their written text and/or social media activity, such as preferences and connections [1, 2]. The applications of stance detection are diverse and encompass domains such as politics, marketing, and social media analysis. Stance detection can be seen as a closely related problem to *sentiment analysis*, also known as *opinion mining* [3, 4]. Sentiment analysis primarily focuses on identifying the explicit sentiment polarity conveyed by a text, typically categorized as Positive, Negative, or Neutral. In contrast, stance detection aims to classify the viewpoint of a given text towards a specific target as Favor, Against, or None. Moreover, the target in stance detection is frequently of an abstract nature, such as ideological topics, and may not be explicitly referenced in the text, while sentiment analysis primarily deals with non-ideological subjects. In addition, the alignment between sentiment and stance within a given text exhibits variability. Consequently, a text may demonstrate positive sentiment while maintaining a stance against the target, or vice versa. To provide an illustration of the stance detection task, Table 1 presents two tweet examples showcasing contrasting stance and sentiment labels. Hashtags in the examples, denoted by the '#' symbol, are used in social media to categorize content and facilitate topic identification.

Table 1 Example of stance detection

Text	**Target**	**Stance**	**Sentiment**
Republicans in the White House will make America great again! #Trump #educateyourself	Hillary Clinton	Against	Positive
And an even worse place from which to make medical decisions FOR OTHER PEOPLE #mybodymychoice #notyours #notgovt	Legalization of Abortion	Favor	Negative

Stance detection poses significant challenges due to its subjective nature, where determining an individual's stance can be highly influenced by personal perspectives. Furthermore, the formation of concepts and opinions involves diverse expressions and linguistic compositions, adding to the difficulty of detection. Particularly in the realm of social media, stance detection becomes even more demanding. Social media text is characterized by brevity, with limitations on character count (e.g., tweets limited to a maximum of 280 characters), extensive use of abbreviations, informality, and inconsistent grammar usage. Additionally, social media discussions tend to be fragmented and lack contextual information, further adding to the challenges faced in stance detection [1, 5].

Previous studies on stance detection have primarily focused on a per-target strategy, where separate models are trained for each target pair and evaluated on test data. Furthermore, previous studies have mainly concentrated on training models solely for stance detection, without incorporating other auxiliary tasks. However, there is potential for enhancing stance detection models by adopting a multi-task learning (MTL) approach. MTL involves training a single model to perform multiple tasks simultaneously, sharing information between them to improve overall performance. MTL has been successful in various machine learning applications, offering advantages like reduced data requirements and improved generalization [6–8].

According to the identified gap in a recent Systematic Literature Review on stance detection by [2], further exploration is required in the field to investigate the potential of developing a joint neural architecture based on the MTL paradigm. In addition, hypotheses regarding the interaction between sentiment and stance appear to be debatable. Several studies have demonstrated a positive interaction between stance and sentiment [9–12], while others have demonstrated that sentiment is inefficient for stance detection models [13–15]. Regarding the sarcasm feature, no study, to the best of our knowledge, has considered sarcasm features for stance detection. In addition, the authors in [16] stated that the errors were mostly in texts that contained sarcastic comments.

Inspired by recent achievements in MTL and the aforementioned research gap, we propose two MTL models to incorporate three interrelated tasks: stance, sentiment, and sarcasm. The proposed models include different schemes for task weighting, with the aim of improving target-specific stance detection. As demonstrated by previous studies, MTL has shown promise in improving the performance of various machine learning techniques [6, 17, 18]. MTL, however, typically involves more complex task management and data preprocessing compared to single-task learning approaches. Furthermore, the implementation of MTL with Transformers is not straightforward, as it is in other deep learning architectures [19]. Hence, our goal is to make the process of building MTL models as simple as building single-task learning models. Moreover, we propose employing diverse approaches for task weighting to assess their impact on the performance of our MTL models. Ultimately, our objective is to enhance the performance of the primary task, stance detection, within an MTL framework by considering and evaluating various weighting schemes that account for the related tasks of sentiment classification and sarcasm detection.

The main contributions of our work can be summarized as follows:

- We introduce two MTL models, namely PMTL and SMTL, which effectively enhance stance detection through the incorporation of sentiment analysis and sarcasm detection tasks.
- We propose the utilization of four task weighting techniques and provide empirical evidence showcasing the effective application of task weighting in MTL models.
- A comprehensive evaluation and analysis are conducted to compare different combinations of the two proposed models, accompanied by various task weighting schemes. Additionally, we demonstrate the advantages of developing a multi-target model in contrast to specific-target models. This evaluation encompasses a thorough assessment of the models on two benchmark datasets in both English and Arabic.
- The experimental results obtained from the evaluation validate the advantages of the proposed models in stance detection. Our most proficient model, a multi-target sequential MTL model with hierarchal weighting (SMTL-HW), achieves state-of-the-art results and surpasses several strong baselines.

The remaining sections of this paper are structured as follows: in Section 2, we provide a comprehensive review of existing literature, evaluating previous research and identifying the gaps that our study seeks to fill. Section 3 elaborates on the methodology employed in our research, outlining the framework and approach used to train our models. The central focus of our study lies in Section 4, where we present our experiments and their results, accompanied by in-depth discussions and interpretations. Lastly, in Section 5, we summarize our significant findings and propose future research directions.

2 Related Work

Stance detection, also known as stance classification and stance prediction, is a relatively new computational problem in the field of social computing. Despite its recent emergence, there has been a noteworthy endeavor to construct models specifically tailored for tackling stance detection [2]. Past studies on stance detection utilized feature engineering with a support vector machine (SVM) classifier [15, 20–23], gradient boosting [24], and k-nearest neighbors (KNN) [25]. Nevertheless, these conventional ML techniques fail to take into account the contextual meaning of words, resulting in relatively lower performance compared to other approaches. Several researchers have subsequently proposed supervised models for stance detection by employing deep learning architectures, including recurrent neural networks (RNNs) [26–28], long short-term memory (LSTM) [29–31], gated recurrent unit (GRU) [12, 32, 33], and convolutional neural network (CNN) [34–37]. Supervised learning-based models excel in accuracy and reliability when paired with suitable algorithms and data representation. However, these models necessitate an ample supply of annotated data that is tailored to the specific task at hand. Obtaining such data can prove challenging in real-world NLP problems due to the vast language diversity and complexity involved. Consequently, this lack of appropriate annotated data can result in failures of supervised learning within these scenarios.

Recently, the field of transfer learning in NLP has witnessed a revolution with the emergence of pre-trained language models like OpenAI GPT [38], Google AI's BERT [39], ELECTRA [40], and T5 [41]. Transfer learning is the process of leveraging knowledge from related domains, tasks, or languages by maximizing the use of unlabeled data in either the source or target domain [42]. Several researchers in the field of stance detection have embraced transfer learning by utilizing pre-trained language models trained on extensive unlabeled data, followed by fine-tuning the models for the classification task. This approach is widely employed in stance detection for domain adaptation [43–45] and cross-lingual learning [46, 47]. In domain adaptation, the documents share the same language but differ in terms of domain or target. For example, source documents might pertain to political tweets, while target documents focus on social issues. On the other hand, in cross-lingual learning, the documents in the source and target domains are written in two different languages, resulting in differing feature spaces. While domain adaptation and cross-lingual learning are effective methods to address the issue of data scarcity and domain shift, they do have some limitations. For instance, in domain adaptation, there is a challenge of finding a suitable source domain and a risk of losing information during adaption. In addition, several factors can impair cross-lingual learning, including differences in language structures, and limited parallel data availability.

Multi-task learning (MTL) is a specific type of transfer learning where a model is trained on multiple tasks simultaneously. A growing body of research on MTL is emerging, especially in the deep learning era. As well as being widely used in computer vision, speech recognition, and recommendation systems; it is being used recently in NLP [6, 17, 18]. In the NLP field, MTL can jointly solve related problems to work towards more general language understanding [8]. This approach has been shown to be effective in a wide range of NLP tasks, such as language translation, sentiment analysis, and text summarization.

In the field of stance detection, Fang et al. [48] were the first to apply MTL by incorporating multiple NLP tasks, including question answering, textual entailment, sentiment analysis, and paraphrase detection. Their MTL model achieved a 14.4% higher macro-F1 score than state-of-the-art (SOTA) models on the FNC-1 dataset [49]. Only a few studies have investigated the joint learning of stance and sentiment detection through MTL [10, 11, 50, 51]. Sun et al. [10] proposed an LSTM-based model to simultaneously capture the stance and sentiment information of a post; however, their model does not incorporate an attention mechanism. The absence of an attention mechanism in text classification models hampers their ability to capture contextual relationships, focus on important information, handle ambiguity, and provide interpretability, resulting in suboptimal performance and reduced transparency. Later, Li et al. [11] enhanced the MTL model by introducing an attention mechanism, along with sentiment and stance features, resulting in improved performance on the SemEval-16 dataset.

Fu et al. [51] addressed the limitation of relying solely on sentiment information for stance detection by introducing an MTL model that integrated opinion-towards classification as another auxiliary task. Other studies have proposed MTL models to jointly address stance detection and rumor veracity prediction, demonstrating their effectiveness [52–58]. Additional studies proposed MTL models trained on multiple targets, treating detecting stances toward N targets as a set of N tasks [59–61]. However, the reported results by

Sobhani et al. [61] show that their proposed single-task attention-based model was more effective than an MTL model trained on multi-targets [62].

The interaction between sentiment and stance is debatable, as some studies suggested a strong relation [9–12], while others found that using sentiment as a feature is inefficient [13–15]. Additionally, no study considers other social dimensions, including sarcasm, and emotion detection. Furthermore, the existing studies on MTL for stance detection have not taken into account task weighting, which can affect the overall performance of the model.

Task weighting is a crucial factor in MTL models. In a multi-task setting, the relative importance of each task can vary, and the model must be able to reflect this in its predictions. This can be achieved through the use of task weights, which reflect the relative importance of each task to the overall objective.

Task weighting approaches can be categorized as equal, proportional, and learning weighting. *Equal weighting* assigns the same weight to each task loss. In the domain of stance detection, all existing studies that have proposed MTL models have uniformly adopted equal weighting for the associated tasks [10, 11, 48, 50, 51, 59, 60]. While this approach is straightforward, it operates under the assumption that all tasks bear equal importance, an assumption that does not hold true for various NLP tasks. In contrast, *proportional weighting* assigns weights to each task loss in proportion to their relative importance. This can be done by using heuristics or domain knowledge to manually assign weights to each task [63, 64]. Lastly, *learning weighting* is an advanced approach in which the optimal weight for each task loss is determined during the training process. This is achieved by minimizing a loss function that combines losses from all tasks, with task-specific weights treated as variables. Through the optimization process, the model can determine the optimal weights that minimize the overall loss. The learning weighting approach has been employed in several studies within the NLP field [65, 66]. To the best of our knowledge, neither proportional nor learning weighting has been proposed specifically for stance detection tasks. Additionally, more research is needed to evaluate the effectiveness of different task weighting schemes on different types of datasets and to assess their generalizability to different domains.

3 Methodology

This section outlines the framework of the proposed MTL models in this study. Two models, namely Parallel Multi-Task Learning (PMTL) and Sequential Multi-Task Learning (SMTL), were introduced, and each model incorporates four distinct task weighting schemes. The proposed models -PMTL and SMTL- are trained to simultaneously predict three tasks: stance, sentiment, and sarcasm. While the primary focus lies on stance detection, the auxiliary tasks of sentiment analysis and sarcasm detection augment the model's comprehension of textual data, consequently enhancing its performance on the primary task. The training process is validated with emphasis on the primary task of stance detection. Figure 1 shows a high-level flow of the proposed MTL models.

The backbone of the proposed MTL models involves fine-tuning the AraBERT-twitter model [67], which encodes both tweets and targets as hidden representations. Hugging Face's Transformers, a library for training BERT-based models, currently supports single-task models, but not modular task heads. Therefore, we implemented a hard parameter sharing [6] where all the tasks share a set of hidden layers, and each task has its output layers (which we call *task head*). Using this approach, the model can learn a shared feature representation that supports the modeling of all tasks.

As shown in Figure 1, the proposed model reflects a pipeline of three components: input layers, shared layers, and task-specific layers. The input and shared layers are the same in both the PMTL and SMTL models, while the layers for task-specific information differ. The subsequent subsections provide a detailed description of each component, followed by an explanation of the task weighting methods.

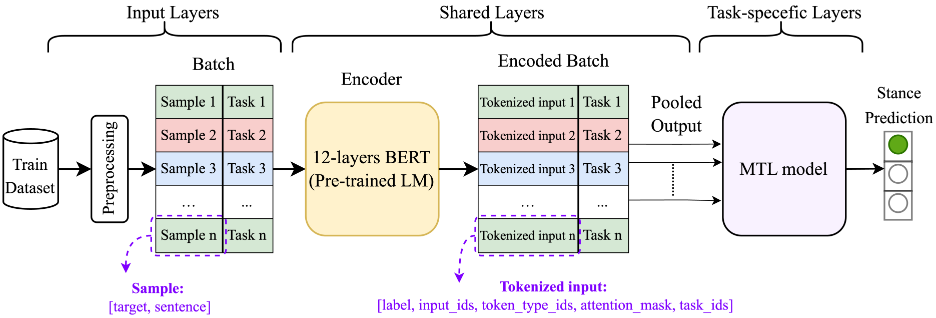

Fig. 1 Illustration of the proposed MTL framework.

3.1 Input Representation

In order to better describe the proposed model, we first introduce some notations. Let $\{D_T\}_{t=1}^{T}$ be data from tasks set, where T is the total number of tasks, and D_t is the training data for task t. Specifically, $D_t = (x_i, y_i)_{i=1}^{N}$ is a set of N examples and the corresponding stance, sentiment, and sarcasm labels. Where x_i denotes the input text and y_i represents the label set for x_i. Table 2 presents definitions of symbols used throughout the description of the proposed framework.

The proposed pipeline starts by preprocessing the input texts (x_i) which involves the removal of URLs, user mentions, extra white spaces, and line breaks. For Arabic texts, an additional preprocessing step is performed, which entails the removal of diacritics, tatweel, and non-Arabic letters. Then, the

Table 2 Symbols definitions.

Symbol	**Explanation**
T	Total number of tasks $t = (1, ..., T)$”
D_t	Training data for task t
N	Number of examples in D_t
x	Input text $x = (x_1, ..., x_N)$
y_i	Label set for x_i
Z_t	Task descriptor generated in the shared layers
$\mathcal{L}_t$	Cross-entropy loss for the task t
θ^{sh}	Shared parameters during the encoding stage
θ^t	Task-specific parameters for output decoder heads

input text is tokenized using a WordPiece tokenizer [68], which splits the text (tweets) into tokens compatible with BERT-based models. Tokenization allows for the generation of word vectors and effectively handles the issue of out-of-vocabulary (OOV) words by splitting them into root words and sub-words.

After completing the data preprocessing and tokenization steps, the multi-task dataset was created by combining samples from three task-specific datasets (i.e., stance, sentiment, and sarcasm). Each sample of the multi-task dataset consists of text, label, task type, and task id. The task type of all three tasks was set to *seq_classification* type as they are sentence classification tasks. In addition, the task id of each sample was added as a new token called *task_ids*, this id will be used by the model to process the samples from each task properly.

3.2 Input Encoder and Task Descriptor

The shared layers are the second component of our proposed MTL framework which allow the model to learn shared representations for each token in the input. These shared representations are subsequently leveraged by the task-specific layers to enhance the model’s performance on each respective task. The shared layers consist of two modules: a shared encoder and a dictionary for the individual task models (task descriptor).

The shared encoder takes in a tokenized input from the input layers and transforms it into three representations: token embeddings, segment embeddings, and position embeddings. These three representations are then element-wise added together to generate a unified representation. This unified representation, of size 128 × 768, is subsequently fed into the large pre-trained language model (BERT in our case) to be fine-tuned. During the fine-tuning stage, the learned contextual embeddings are applied to individual tasks to accommodate the multi-task setting. Additionally, a task-specific dictionary is created, which includes the encoded input and the task descriptor Z_t, a label that identifies the task that the model is currently working on. The dictionary is passed to the task-specific layers, which are responsible for predicting the output of the task, as explained in the following section.

3.3 Multi-task Learning Model

This section describes the main component of our proposed framework, namely the task-specific layers that constitute the MTL model. In particular, we proposed two models, namely PMTL and SMTL. PMTL involves training multiple tasks simultaneously with each task having its own set of parameters, while SMTL trains tasks simultaneously and sequentially where the knowledge learned from earlier tasks is transferred when training subsequent tasks. Figure 2 shows a high-level flow of the proposed PMTL and SMTL models.

These task-specific layers define the MTL objective [1] by jointly minimizing the loss of each task $\mathcal{L}_t$ as follows:

$$obj(MTL) = min_{\theta^{sh},\theta^{1},..,\theta^{T}} \sum_{t=1}^{T} \mathcal{L}_t \left(\left\{ \theta^{sh}, \theta^{T} \right\}, D_t \right) \tag{1}$$

[1] For clarity, we use the term “MTL objective” to refer to the final learning objective of a model, while “loss” represents an individual component within this objective function.

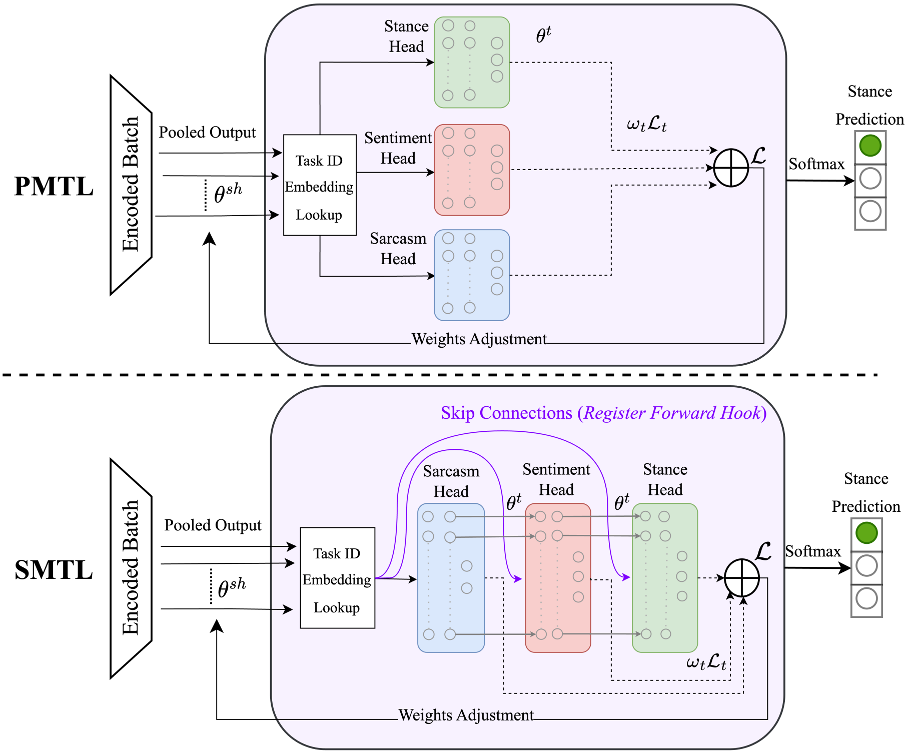

Fig. 2 Illustration of our proposed PMTL and SMTL models.

where $\mathcal{L}_t$ is the cross-entropy loss for the task t. The objective of this loss is to measure the similarity between the probability distribution generated by the Softmax function and the actual category distribution. Specifically, it penalizes wrong predictions by optimizing the negative log-likelihood of the correct prediction. The shared learnable generated weights θ^{sh} are the weights learned by the shared encoder during the previous encoding stage, and the task-specific learnable generated weights θ^t are the weights learned by the task-specific decoder heads.

Figure 3 simplifies the formulation of the problem and visually represents the distinction between the typical single-task model and our proposed multi-task model. As shown in this figure, our MTL model f_θ can be defined as follows:

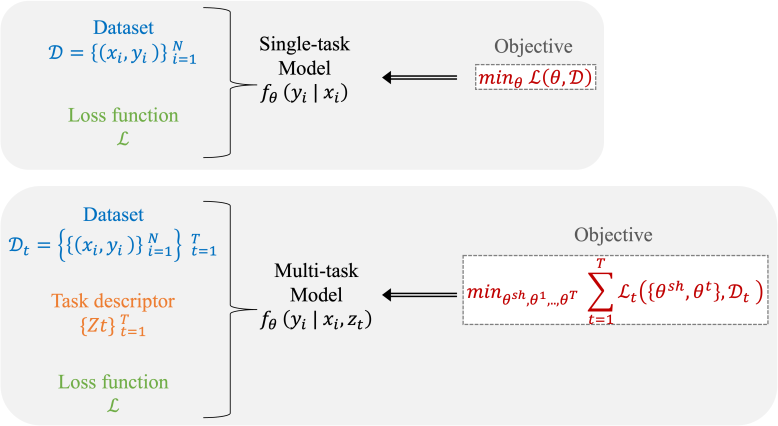
Dataset
$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
Loss function
$\mathcal{L}$
Single-task
Model
$f_\theta(y_i \mid x_i)$
Objective
$min_\theta \, \mathcal{L}(\theta, \mathcal{D})$
Dataset
$\mathcal{D}_t = \left\{\{(x_i, y_i)\}_{i=1}^{N}\right\}_{t=1}^{T}$
Task descriptor
$\{Zt\}_{t=1}^{T}$
Loss function
$\mathcal{L}$
Multi-task
Model
$f_\theta(y_i \mid x_i, z_t)$
Objective
$min_{\theta^{sh}, \theta^1, ..., \theta^T} \sum_{t=1}^{T} \mathcal{L}_t(\{\theta^{sh}, \theta^t\}, \mathcal{D}_t)$

Fig. 3 Simplification of the problem formulation in a single-task model versus the proposed multi-task model. The donations are presented in Table 2.

$$f_\theta(y_i \mid x_i, z_t) = obj(MTL) \tag{2}$$

where x_i is the input text and y_i is the label set for x_i from a given training data D_t for task t. The label set is varied based on the selected task (i.e., stance, sentiment, or sarcasm). Therefore, the model predicts the label y_i given the embeddings of the input x_i and the task descriptor Z_t generated by the shared layers.

As previously mentioned, the proposed models comprise three components: input layers, shared layers, and task-specific layers. While the input and shared layers remain consistent in both PMTL and SMTL models, the layers associated with task-specific information differ. Figure 2 visually represents these distinctions. As shown in this figure, in PMTL, all tasks are simultaneously and independently learned. Conversely, in SMTL, the tasks are sequentially learned, enabling the target task (i.e., stance) to capitalize on the features acquired from the source tasks (i.e., sarcasm and sentiment). In the following, we elaborate on the difference between PMTL and SMTL models, and then we describe the objective of the SMTL model.

PMTL and SMTL can be seen as being on different time intervals. Assume we have two tasks; task $T1$ trained during the interval $\{t_1, t_2\}$, and task $T2$ trained during the interval $\{t_3, t_4\}$. In the PMTL setting, $t_1 = t_3$ and $t_2 = t_4$. That means, training commences and concludes simultaneously in both tasks. However, in STML, the second task is trained after training of the first task has started, where $t_1 < t_3$ and $t_3 < t_4$. Furthermore, a main characteristic of SMTL is that the features learned in the source task-specific layers are transferred to the target layers. Meanwhile, the task-specific layers are not shared between the different tasks in the PMTL paradigm. Figure 4 illustrates the difference in the training intervals between the two paradigms.

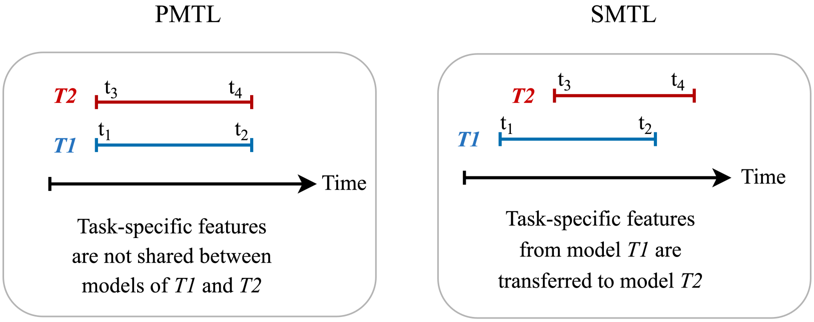

Fig. 4 Training intervals of two tasks *T1* and *T2* in PMTL versus SMTL.

The proposed SMTL model can inherently avoid the *catastrophic forgetting*, a common problem for sequential transfer learning. Catastrophic forgetting occurs when a model overfits the target domain, forgetting previously learned knowledge from the source tasks [69]. To overcome this problem, we integrate the idea of MTL into sequential transfer learning. In particular, our SMTL model is designed to fulfill three main objectives. Firstly, it is trained on a comprehensive dataset that encompasses examples from all tasks, enabling simultaneous learning and prediction for multiple tasks. Secondly, it aims to minimize the loss of the target task along with the losses of the source tasks. This objective shared similarities with the PMTL objective (presented in Equation 1) but distinguished itself by consistently including the loss of the source tasks in the optimization objective to prevent catastrophic forgetting. Lastly, to facilitate sequential knowledge transfer, skip connections were integrated to extract ”features” from the source models instead of ”class logits.” To establish these connections and track the generated features and losses, an identity operator layer was introduced, ensuring the input passed through without alteration. The implementation of this mechanism involved employing a *register forward hook* function, which registered a global forward hook for all sub-models and was invoked after the ”forward” function generated a hidden representation or computed an output (see Figure 2).

3.4 Task Weighting

In the context of MTL, assigning appropriate task weights is crucial to ensure that the relative importance of each task is accurately reflected. The task weights should be carefully calibrated to strike a balance between optimizing the performance of the main task and considering the contributions of related tasks. It is worth noting that different tasks can have different objectives, and the task-specific loss function might differ based on the task. For instance, classification problems often employ cross-entropy loss, while regression problems usually utilize mean squared error. In the subsequent sections, we provide a comprehensive explanation of the MTL objective incorporating task weighting approaches. We then provide a detailed description of the various task weighting methods that have been employed in our proposed models.

3.4.1 Training Objective

When implementing MTL models, it is common for the tasks included to compete with each other. A phenomenon, known as *task imbalance*, occurs when we are unable to appropriately balance these tasks [65]. In the context of MTL settings, it is essential to establish both a loss function and an optimizer to effectively train the deep learning model. The MTL loss function is typically a combination of multiple loss functions, corresponding to multiple tasks involved in the model training. If one loss is much larger than the others, then its corresponding task may *dominate* the training. In addition, some losses may converge faster or might be more important to the overall system objective. Furthermore, the optimization method is not aware of each individual task loss; thus, performance in MTL-based models is greatly influenced by the relative weights assigned to each task. For example, when all tasks except one are set to zero, then only that task will be optimized.

In our model, our primary focus is to prioritize the stance detection task during the training process, while considering sentiment classification and sarcasm detection as auxiliary tasks. To achieve this, we modify the MTL objective function, presented in Equation 1, by introducing a task importance coefficient, as follows:

$$obj(MTL) = min_{\theta^{sh},\theta^{1},..,\theta^{T}} \sum_{t=1}^{T} \omega_t \mathcal{L}_t \left(\left\{ \theta^{sh}, \theta^{T} \right\}, D_t \right) \tag{3}$$

where ω_t denotes the importance coefficient (i.e., weight) for task t. The assignment of appropriate weights to each task's loss is of utmost importance. The simplest method is to set them equally, i.e., $\omega = \frac{1}{T}$. It is common, however, to view weights as hyper-parameters that are set based on grid search or experience. Besides, weight adaptation methods formulate the MTL optimization problem by adaptively adjusting the weights of the tasks during training in accordance with a predefined heuristic. In the following section, we describe the different weighting schemes that we proposed for our MTL models.

3.4.2 Weighting Methods

In our experimental study, we introduce various weighting schemes to investigate their impact on MTL models. Specifically, we propose the inclusion of the following four loss-balanced task weighting schemes in the MTL objective:

- **Static weighted sum (SW)**: In this approach, we assign a fixed weight to each task, which determines the importance coefficient of the respective task. Denoting the stance loss as ($\mathcal{L}_{st}$), sentiment loss as ($\mathcal{L}_{sen}$), and sarcasm loss as ($\mathcal{L}_{sar}$); the overall loss ($\mathcal{L}$) in the MTL optimization objective is defined as:

  $$\mathcal{L} = \omega_1 \mathcal{L}_{st} + \omega_2 \mathcal{L}_{sen} + \omega_3 \mathcal{L}_{sar} \tag{4}$$

  where ω_1, ω_2, and ω_3 control the weight of $\mathcal{L}_{st}$, $\mathcal{L}_{sen}$, and $\mathcal{L}_{sar}$, respectively.

  According to our empirical analysis, setting $\omega_1 = 0.6$, $\omega_2 = 0.3$, and $\omega_3 = 0.1$ results in the best performance of our models. This suggests that the stance detection task is considered more crucial or has a higher impact on the overall objective of the MTL model. Furthermore, the sentiment analysis task was given more weight compared to the sarcasm task, possibly because sentiment analysis is deemed more relevant or informative in the context of stance detection.

- **Relative weighted sum (RW)**: Based on the intuition that tasks with higher training loss should receive more attention, we propose a dynamic weight assignment strategy that assigns a larger weight to the stance loss $\mathcal{L}_{st}$ during the optimization process. We infer the loss weightings by observing the loss values during model training. The overall loss $\mathcal{L}$ in the MTL optimization objective, incorporating the RW technique, is defined as:

$$\mathcal{L} = \omega\mathcal{L}_{st} + \frac{\omega}{2}\mathcal{L}_{sen} + \frac{\omega}{3}\mathcal{L}_{sar} \tag{5}$$

Here, we train the network to learn a single parameter ω, which serves as the weight for the stance detection task. We assign ω to prioritize the stance detection task, while *relatively* assigning smaller weights to the sentiment and sarcasm tasks.

- **Hierarchical weighting (HW)**: This is a dynamic weight assignment strategy that assigns a larger weight to the lower-level tasks (i.e., sentiment and sarcasm) during the early stages of training, and then assigns a larger weight to the target task (i.e., stance) during the later stages of the training. This is based on the assumption that the model should focus on learning the lower-level tasks first, as these tasks are necessary for learning the target task. For example, the sentiment task is necessary for learning the stance task, as the stance of a text is often related to its sentiment. In this approach, $\mathcal{L}$ is defined as:

$$\mathcal{L} = \omega\mathcal{L}_{st} + \mathcal{L}_{sen} + \mathcal{L}_{sar} \tag{6}$$

where the learnable generated weight ω is dynamically updated as follows:

$$\omega = max\left(min\left(\frac{\mathcal{L}_{st}}{\mathcal{L}_{sen}}.\omega, 2\right), 1\right) \tag{7}$$

The weight, ω, is utilized to regulate the relative significance of $\mathcal{L}_{st}$ based on empirical assumptions that $\mathcal{L}_{sen}$ and $\mathcal{L}_{sar}$ carry equal importance. Initially set to 1, ω ensures equal emphasis on optimizing all tasks until $\mathcal{L}_{sen}$ becomes relatively smaller than $\mathcal{L}_{st}$. Consequently, as $\mathcal{L}_{st}$ increases, the model will progressively focus more on the stance detection task.

- **Uncertainty weighting (UW)**: This approach is grounded on the notion that tasks with higher uncertainty should be assigned lower weights compared to tasks with lower uncertainty. Following [70], we employed *homoscedastic* uncertainty for task weighting. In this approach, the overall loss $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \sum_{t=1}^{T} \frac{1}{{\sigma_t}^2} \mathcal{L}_t + \log \sigma_t \tag{8}$$

where σ_t is the homoscedastic uncertainty associated with each task. As a practical matter, we train the network to learn the log-variance, $log{\sigma_t}^2$, since it is more numerically stable than ${\sigma_t}^2$ as $\mathcal{L}_t$ avoids any division by zero.

It is evident from equation 8 that the increase in uncertainty value will result in a smaller contribution of the task to the overall loss (i.e. if σ_t increases, the weight of $\mathcal{L}_t$ decreases). The second term, $\log \sigma_t$, acts as a regularization term to prevent the model from learning a trivial solution by setting the uncertainty of all tasks (i.e., σ_t) to extremely high value.

In [70], the authors propose using homoscedastic uncertainty, a task-specific uncertainty that remains constant for different input data. Homoscedastic uncertainty arises when tasks exhibit comparable difficulty levels, resulting in consistent model performance and consistent uncertainty or error across all tasks. In their work [70], the authors show that this approach outperforms the naive approach (i.e., the weighted linear sum of the losses) in the context of visual scene understanding, which includes scene geometry and semantics. While their work primarily focuses on regression, we adapt their formulation for a classification problem. Equation 8 presents a simplified version of the derived MTL loss, with a comprehensive derivation available in [70].

4 Experiments

4.1 Datasets

The primary focus of this study is to propose a stance detection model specifically designed for the Arabic language, utilizing the Mawqif dataset [71]. However, in order to evaluate the generalizability of our model, we also conducted experiments on the SemEval-16 dataset [72], which is an English dataset widely used for stance detection. Both datasets consist of Twitter posts that have been annotated with stance and sentiment labels. Notably, the Mawqif dataset also includes annotations for sarcasm, providing additional valuable information for our proposed model.

The Mawqif dataset [71] stands as the pioneering and sole dataset made available to facilitate research and development of target-specific stance detection models in the Arabic language. The dataset comprises 4,121 tweets written in multiple dialects of Arabic and focusing on three topics: "women empowerment," "COVID-19 vaccine," and "digital transformation.". Each tweet is assigned a target and manually annotated with stance, sentiment, and sarcasm polarities. The stance annotations are ternary, indicating whether the stance of a tweet towards a specific target is in *favor*, *against*, or *none* if the text does not provide sufficient stance information. The sentiment annotations are also ternary, indicating whether the tweet is *positive*, *negative*, or *neutral*. The sarcasm annotations are binary, indicating whether the tweet is *sarcastic*, or *non-sarcastic*.

The SemEval-16 dataset [72] is an English dataset for stance detection, which was first introduced in 2016 as part of a shared task. Furthermore, it has been widely used as a benchmark for stance detection research and has been the basis for several ML models. The dataset consists of 4,163 tweets manually annotated with a stance label (*favor*, *against*, or *none*), as well as a sentiment label (*positive*, *negative*, or *neutral*). The dataset was collected during the 2016 US presidential election campaign and it covers five targets: “Atheism”, “Climate Change”, “Feminist Movement”, “Hillary Clinton”, and “Legalization of Abortion”. The detailed statistics of both MAWQIF and SEMEVAL-16 datasets are listed in Table 3.

Table 3 Data distribution of MAWQIF dataset and SEMEVAL-16 dataset.

Dataset	**Target**	#Train	%Favor	%Against	%None	#Test	%Favor	%Against	%None
Mawqif	Covid vaccine	1167	43.62	43.53	12.85	206	43.69	43.69	12.62
	Digital trans.	1145	76.77	12.40	10.83	203	76.85	12.32	10.84
	Women emp.	1190	63.87	31.18	4.96	210	63.81	30.95	5.24
	Total	**3502**	**61.34**	**29.15**	**9.51**	**619**	**61.39**	**29.08**	**9.53**
SemEval	Athesim	513	17.9	59.3	22.8	220	14.5	72.7	12.7
	Climate change	395	53.7	3.8	42.5	169	72.8	6.5	20.7
	Feminism	664	31.6	49.4	19	285	20.4	64.2	15.4
	Hillary Clinton	689	17.1	57	25.8	295	15.3	58.3	26.4
	Abortion	653	18.5	54.4	27.1	280	16.4	67.5	16.1
	Total	**2914**	**25.8**	**47.9**	**26.3**	**1249**	**23.1**	**51.8**	**25.1**

4.2 Experimental Setup

Model Configuration. As an integral component in our proposed PMTL and SMTL models, we conducted fine-tuning on the AraBERT-twitter model [67]. This process involves encoding both tweets and targets as hidden representations. The resultant model serves as the backbone model for training the Mwqif dataset. In a similar vein, for the SEMEVAL-16 dataset, we performed fine-tuning on RoBERTa [73], leveraging hidden representations that encode both tweets and targets.

All experiments were run on a single NVIDIA GeForce RTX 3090, 24 GB. We set the maximum sequence length of the input to 128 tokens, a feature dimension to 786, and the batch size to 32. Each of the models was fine-tuned for 20 epochs with a dropout rate of 0.1. We set a hyper-parameter, known as "patience" to 5, which denotes the number of epochs without improvement after which training will be stopped. Adam with decoupled Weight decay (AdamW) [74] was selected for optimization with a learning rate of 2e-5. Compared to the Adam optimizer [75], the AdamW optimizer has better generalizability and results in a lower training loss [74]. To prevent overfitting, we set weight decay to 1e-5. All experiments were performed with a fixed initialization seed by setting Pytorch global seed to 42. The hyper-parameters were selected empirically in these experiments. Table 4 summarizes the hyper-parameters values used in our experiments.

Table 4 Hyper-parameter values.

Hyper-parameter	**Value**
Max. sequence length	128
Feature dimension	768
Batch size	32
Number of epochs	20
Dropout rate	0.1
Early stop patience	5
Optimizer	AdamW
Learning rate	2e-5
weight decay	1e-5

As shown in Table 3, both datasets are split into training and testing sets. For all experiments, we further split off 15% of the training set for model development. It should be mentioned that we tuned our model only on the development set. The model's performance on the test set is then used as a proxy for its ability to generalize to new inputs.

Evaluation Metrics. Our models were evaluated using the macro-average F1 (F_{Mac}) and the micro-average F1 (F_{Mic}) to align with previous stance detection studies that report their results using these metrics [72]. First, the F1-score is computed for the "Favor" and "Against" classes as follows:

$$F_{favor} = \frac{2P_{favor}R_{favor}}{P_{favor} + R_{favor}} \tag{9}$$

$$F_{against} = \frac{2P_{against}R_{against}}{P_{against} + R_{against}} \tag{10}$$

where P and R denote for precision and recall, respectively. Then, F_{Mac} is calculated for each target as follows:

$$F_{Mac} = \frac{F_{favor} + F_{against}}{2} \tag{11}$$

Note that the "*none*" class, a class that was scarcely in the data, was not disregarded during training. However, this class was not considered in the evaluation because we are only interested in "Favor" and "Against" labels in this task. This approach is consistent with other stance detection studies, where reporting results using F_{Mac} specifically for the "favor" and "against" stance labels is a common practice.

By averaging the individual F_{Mac} scores calculated for each target, we obtain the F_{Mac} across targets. This metric provides an overall performance measure that takes into account imbalanced data, ensuring equal contribution from both majority and minority classes. Additionally, we report our results using the F_{Mic} metric, which involves computing F_{favor} and $F_{against}$ scores across all targets and then taking their average. This measure is particularly useful for models performing well on more frequent target classes. However, achieving a high F_{Mac} score requires the model to perform well across all target classes.

4.3 Results and Discussion

In this section, we will present and compare the performance of the proposed models. The objective is to identify the most effective approach for target-specific stance detection, considering both the Mawqif and SemEval-16 datasets. In addition, we will discuss and analyze the results in three dimensions. First, we will discuss the performance of the two multi-task model models and compare them with the single-task model. Second, we will analyze the effect of task weighting on the performance of the models. Third, we will evaluate the performance of the multi-target classifier compared to a target-specific classifier. Additionally, we will provide an attention visualization to gain insights into which parts of the input text the models are paying more attention to when making their predictions. It should be noted that our analysis is performed on the test set. The outcomes of this section will guide the selection of the best approach for stance detection.

Before presenting the results, it is important to provide definitions of all the model variations that were proposed; these model variations are as follows:

- PMTL: Parallel Multi-Task Learning model that leverages three tasks: stance, sentiment, and sarcasm. This model is illustrated in Figure 2.
- PMTL-sent: PMTL model that leverages two tasks: stance and sentiment.
- PMTL-sarc: PMTL model that leverages two tasks: stance and sarcasm.
- PMTL-SW: Best PMTL setting with static weighted loss.
- PMTL-RW: Best PMTL setting with relative weighted loss.
- PMTL-HW: Best PMTL setting with hierarchical weighted loss.
- PMTL-UW: Best PMTL setting with uncertainty weighted loss.
- SMTL-sarc-sent: Sequential Multi-Task Learning model that trains three tasks in the following order: sarcasm, sentiment, and stance. This model is illustrated in Figure 2.

- SMTL-sent-sarc: SMTL trains three tasks in the following order: sentiment, sarcasm, and stance.
- SMTL-sent: SMTL trains two tasks, sentiment followed by stance.
- SMTL-sarc: SMTL trains two tasks, sarcasm followed by stance.
- SMTL-SW: Best SMTL setting with static weighted loss.
- SMTL-RW: Best SMTL setting with relative weighted loss.
- SMTL-HW: Best SMTL setting with hierarchical weighted loss.
- SMTL-UW: Best SMTL setting with uncertainty weighted loss.

4.3.1 Multi-task Model Architecture

To shed light on the effectiveness of MTL in improving the performance of a stance detection task, we compared the performance of the two proposed architectures, PMTL and SMTL, without task weighting. Table 5 presents the performance of all proposed models on Mawqif dataset. The performance is measured in terms of F1-score for the "Favor" and "Against" classes (F_{favor}, $F_{against}$), macro F1-score (F_{Mac}), and micro F1-score (F_{Mic}). For each model variation, we trained three classifiers on each target separately.

To assess the proposed models' generalization capability, we evaluated their performance on SemEval-16 dataset [72], an English dataset. Testing on another language, such as English, will provide a reliable estimate of the model's ability to generalize to new languages. Due to the distinct structural and grammatical differences between the English and Arabic languages, incorporating evaluations using English text aids in evaluating the robustness of the proposed models. The performance of the proposed models on the SemEval-16 dataset is presented in Table 6, presenting the results obtained from training five classifiers individually for each target.

As indicated by the results presented in Tables 5 and 6, the superiority of the SMTL approach over PMTL can be observed in both datasets, namely

Table 5 F1-scores of multi-task models on MAWQIF dataset reported for each individual target. "Overall" reports F1-scores calculated globally across all targets. **Bold** for best within each model group. Green for best among all models and red for second-best.

	COVID-19 Vaccine			Digital Transformation			Women Empowerment			Overall	
	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}	F_{Mic}	F_{Mac}
PMTL models											
PMTL-sent	81.52	81.32	81.42	90.85	56.00	73.43	89.68	80.95	85.32	81.49	80.05
PMTL-sarc	81.82	80.43	81.13	90.18	55.81	73.00	89.45	81.82	85.64	80.19	79.92
PMTL	**82.15**	**82.02**	**82.09**	**91.02**	**65.22**	**78.12**	**90.37**	**86.13**	**88.25**	**82.82**	**81.92**
PMTL + Loss weighting models											
PMTL-SW	82.66	83.05	82.86	**90.96**	**68.38**	**79.67**	**91.91**	**87.22**	**89.56**	**83.61**	**84.03**
PMTL-RW	80.23	80.43	80.33	89.81	63.16	76.48	90.65	81.89	86.27	81.21	81.03
PMTL-HW	**82.44**	**83.61**	**83.02**	90.52	59.09	74.81	91.10	83.87	87.49	81.54	81.77
PMTL-UW	81.71	78.79	80.25	89.85	55.81	72.83	90.58	84.85	87.71	79.51	80.26
SMTL models											
SMTL-sent	79.04	79.38	79.21	89.30	51.16	70.23	91.73	85.51	88.62	79.81	79.35
SMTL-sarc	80.00	80.00	80.00	89.97	56.52	73.25	90.11	80.00	85.05	79.74	79.43
SMTL-sent-sarc	79.01	81.16	80.09	89.46	54.90	72.18	91.45	85.93	88.69	81.06	80.32
SMTL-sarc-sent	**80.92**	**83.51**	**82.22**	**91.13**	**68.09**	**79.61**	**92.00**	**86.11**	**89.06**	**83.02**	**83.63**
SMTL-sarc-sent + Loss weighting models											
SMTL-SW	83.08	84.16	83.62	90.74	56.52	73.63	91.04	84.38	87.71	81.28	81.65
SMTL-RW	76.83	78.00	77.41	89.70	54.55	72.12	90.18	83.08	86.63	79.20	78.72
SMTL-HW	**83.50**	**85.82**	**84.66**	**92.30**	**68.64**	**80.47**	**93.32**	**87.00**	**90.16**	**84.01**	**85.10**
SMTL-UW	81.61	82.90	82.26	91.24	63.41	77.33	91.24	85.71	88.48	83.32	82.69

MAWQIF and SEMEVAL-16. This observation holds true regardless of task weighting, which will be further elaborated upon in Section 4.3.2.

Regarding incorporating sentiment and sarcasm tasks in stance detection models, the inclusion of both tasks in PMTL and SMTL confers a significant advantage over models that solely focus on sentiment or sarcasm. As shown in Table 5, PMTL with both sentiment and sarcasm has the highest Macro F1 score of 81.92, which is around 2 points above PMTL-sent and PMTL-sarc. A similar conclusion is found for SMTL, which has the highest Macro F1 score of 83.63 when incorporating both tasks. This is 4 points higher than SMTL-sent

Table 6 F1-scores of multi-task models on SemEval-16 dataset reported for each individual target. "Overall" reports F1-scores calculated globally across all targets. **Bold** for best within each model group. 🟩 Green for best among all models and 🟥 red for second-best.

	Atheism			Climate change			Feminist movement			Hilary Clinton			Abortion legalization			Overall	
	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}	F_{Mic}	F_{Mac}
Multi-task models																	
PMTL	**61.54**	**86.58**	**74.06**	**91.41**	15.38	53.4	52.22	63.48	57.85	47.89	78.17	63.03	52.83	77.14	64.99	72.63	62.66
SMTL	59.15	86.26	72.71	91.27	**16.67**	**53.97**	**53.85**	**67.35**	**60.60**	**52.27**	**80.00**	**66.14**	**57.14**	**78.16**	**67.65**	**73.63**	**64.21**
PMTL + Loss weighting																	
PMTL-SW	**63.16**	**87.09**	**75.13**	**92.98**	**29.38**	**61.18**	**54.44**	73.02	**63.73**	**61.70**	79.65	**70.67**	57.55	**78.53**	**68.04**	**74.83**	**67.75**
PMTL-RW	50.60	80.68	65.64	90.91	28.57	59.74	52.23	**73.62**	62.92	60.00	**80.23**	70.12	**59.32**	76.42	67.87	74.75	65.26
PMTL-HW	57.89	85.71	71.80	91.05	16.67	53.86	54.02	71.20	62.61	56.1	77.38	66.74	54.90	77.81	66.36	73.81	64.27
PMTL-UW	61.11	87.01	74.06	91.34	16.67	54.00	48.31	68.81	58.56	38.24	78.72	58.48	54.21	77.97	66.09	70.83	62.24
SMTL + Loss weighting																	
SMTL-SW	62.79	85.32	74.06	90.98	15.38	53.18	53.99	72.45	63.22	60.87	81.98	71.42	54.39	77.06	65.72	74.79	65.52
SMTL-RW	61.76	**88.10**	74.93	91.54	15.38	53.46	54.44	71.10	62.77	56.10	82.76	69.43	52.17	72.95	62.56	74.14	64.63
SMTL-HW	**67.57**	87.79	**77.68**	**92.55**	**28.57**	**60.56**	**55.00**	**72.51**	**63.76**	**61.95**	**82.35**	**72.15**	**58.85**	**79.34**	**69.10**	**75.42**	**68.65**
SMTL-UW	62.16	86.67	74.41	91.95	15.38	53.67	51.14	71.75	61.44	48.78	78.92	63.85	46.34	77.38	61.86	71.45	63.05

and SMTL-sarc. Remarkably, the SMTL model trained on sarcasm first and then sentiment performed better than a model trained on sentiment and then sarcasm. When the model is trained on sarcasm first, it can potentially use the sentiment understanding it gains from sarcasm detection to improve its ability to identify stance-related sentiments in text. Overall, the results indicate that auxiliary tasks can significantly improve the performance of the main task.

4.3.2 Task Weighting

In this study, we have introduced four task weighting schemes: static weighting (SW), relative weighting (RW), hierarchical weighting (HW), and uncertainty weighting (UW). Section 3.4.2 provides an illustration of these schemes. Our investigation in this section focused on analyzing their impact on the proposed MTL models. The experimental results consistently revealed the positive influence of task weighting on the performance of both PMTL and SMTL models. This improvement was observed across the Mawqif and SemEval-16 datasets, as shown in Tables 5 and 6. Nevertheless, certain weighting schemes are more effective than others, as elucidated in the subsequent paragraphs.

The evaluation results for Mawqif dataset presented in Table 5 demonstrate that SW provides a clear advantage over other weighting schemes for the PMTL model, with a Macro F1 score of 84.03. This is 2 points higher than RW and HW, and 4 points higher than UW. On the other hand, for the SMTL model, HW has the highest overall Macro F1 score of 85.1. This is 3 points higher than UW, 4 points higher than SW, and 6 points higher than RW.

The same conclusion regarding PMTL also applies to SemEval-16 dataset. Table 6 shows that among all PMTL models, PMTL with SW achieved the highest F1 score of 67.75. This is 2 points higher than RW, 3 points higher than HW, and 5 points higher than UW. On the other hand, the SMTL model performed the best when combined with the HW weighting scheme, scoring an F1 of 68.65. This is 3, 4, and 5 points higher than SW, RW, and UW, respectively. Nevertheless, it is important to acknowledge that models relying on learnable weights exhibit slower training in comparison to those utilizing constant parameters.

4.3.3 Multi-target Classifier

Recall that a *multi-target* classifier is a model trained on multiple targets (i.e., topics) simultaneously, whereas a *target-specific* classifier is trained on only one topic. To compare the two model variations, the performance metrics of the single-target classifiers were averaged and reported against the performance of the multi-target classifier. Tables 7 and 8 show the comparison for Mawqif and SemEval-16 datasets, respectively. According to the reported results, combining all targets into a single classifier seems to be a superior solution compared to training separate models for each target. This observation remains consistent for both datasets.

Our results have implications for the development of stance classification models. In particular, our findings suggest that it is beneficial to train models

Table 7 F1-scores of multi-task models on Mawqif dataset for overall target-specific vs. multi-target. **Bold** for best within each model group. Green for best among all models and red for second-best. Underlined for best F_{Mac} comparing between target-specific and multi-target.

	Overall target-specific			**Multi-target**		
	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}
PMTL models						
PMTL-sent	88.13	74.85	80.05	88.38	78.80	83.59
PMTL-sarc	87.22	73.15	79.92	87.61	77.38	82.50
PMTL	**89.70**	**75.13**	**83.03**	**89.09**	**78.98**	**84.03**
PMTL + Loss weighting models						
PMTL-SW	**88.63**	**78.59**	**84.03**	**89.32**	80.22	**84.77**
PMTL-RW	86.99	75.42	81.03	88.57	**80.43**	84.50
PMTL-HW	87.98	75.10	81.77	88.89	79.45	84.17
PMTL-UW	87.04	71.97	80.26	88.71	78.95	83.83
SMTL models						
SMTL-sent	86.90	72.71	79.35	88.17	78.33	83.25
SMTL-sarc	87.00	72.47	79.43	87.92	76.88	82.40
SMTL-sent-sarc	**87.31**	74.81	80.32	88.54	**78.74**	83.64
SMTL-sarc-sent	87.07	**78.98**	**83.63**	**89.74**	78.40	**84.07**
SMTL-sarc-sent + Loss weighting models						
SMTL-SW	88.07	74.50	81.65	88.04	77.84	82.94
SMTL-RW	85.92	72.48	78.72	86.97	78.33	82.65
SMTL-HW	**89.30**	**78.72**	**85.10**	**90.42**	**82.05**	**86.23**
SMTL-UW	88.15	78.48	82.69	87.63	77.95	82.79

on multiple targets, rather than on a single target. This is likely because the multi-target model has access to a much larger amount of data. In addition, a multi-target model can learn to share information between the different targets to identify stances towards all of those targets. Thus, it will be more likely to learn generic stance characteristics rather than particular traits of stance towards a single target.

Furthermore, by examining the results presented in Tables, 7 and 8, we observe that the multi-target SMTL-HW model outperforms others, attaining an F_{Mac} score of 86.23 on the Mawqif dataset and 73.23 on the SemEval-16

Table 8 F1-scores of multi-task models on SemEval-16 dataset for overall target-specific vs. multi-target. **Bold** for best within each model group. Green for best among all models and red for second-best. Underlined for best F_{Mac} comparing between target-specific and multi-target.

	Overall target-specific			**Multi-target**		
	F_{favor}	$F_{against}$	F_{Mac}	F_{favor}	$F_{against}$	F_{Mac}
Multi-task models						
PMTL	70.59	74.67	62.66	58.17	76.92	67.55
SMTL	**71.02**	**76.23**	**64.21**	**62.32**	**79.61**	**70.96**
PMTL + Loss weighting						
PMTL-SW	**72.65**	77.02	**67.75**	**66.44**	80.03	**72.63**
PMTL-RW	72.46	**77.03**	65.26	63.18	**80.91**	72.05
PMTL-HW	71.83	75.79	64.27	63.84	79.08	71.46
PMTL-UW	65.62	76.04	62.24	63.47	75.99	69.73
SMTL + Loss weighting						
SMTL-SW	72.14	77.43	65.52	**67.14**	74.3	70.72
SMTL-RW	71.31	76.96	64.63	66.88	77.59	72.24
SMTL-HW	**73.07**	**77.77**	**68.65**	67.02	**78.96**	**73.23**
SMTL-UW	66.20	76.70	63.05	64.35	77.86	71.10

dataset. Hence, it can be concluded that the multi-target SMTL-HW model demonstrates the highest performance among the evaluated models.

Although both multi-target and single-target models showed good performance, there were some targets that were easier for the models to identify the stances towards. For example, in Mawqif dataset, all models performed best when considering the "women empowerment" target, as shown in Table 5. In the case of SemEval-16 dataset, Table 6 shows that all models performed best when considering the "Atheism" target. These findings suggest that tweets related to women empowerment or atheism may contain strong indicators that differentiate between instances expressing support and those expressing opposition.

4.3.4 Attention Visualizations

As part of our analysis, we explore attention visualizations to offer insights into how our model processes and attends to input text. Specifically, we visualize the attention weights between the [CLS] token and all other tokens in the last layer of the best-performed model, multi-target SMTL-HW, by using LIME [76] method. By examining these attention weights, we can gain a better understanding of which parts of the input text are most important for the model's predictions. It should be mentioned that our analysis was performed on the test set, which allows us to evaluate the generalizability of the model to new and unseen data.

Table 9 shows the attention weights of the last layer in SMTL-HW model for randomly selected input sentences whose labels were accurately predicted by SMTL-HW. In the visualizations, words with darker colors indicate greater significance in influencing the model's predictions. We can observe that SMTL-HW model exhibits the capability to effectively capture prominent entities and sentiments within the text. For instance, in the first sentence, SMTL-HW highlights "alleged" and "capitalist," which are non-trivial terms representing an opposing stance towards women's empowerment. In the second sentence, the model selects the words "compulsion", "die", and "fear" as highly relevant to the topic of the COVID-19 vaccine. Furthermore, the SMTL-HW model identifies words that support the notion of digital transformation, such as "value," "benefit," and "traffic." By attending to these terms, the model demonstrates an understanding of the positive aspects and advantages associated with digitization processes. Overall, the attention visualizations obtained from the SMTL-HW model provide insights into its ability to capture significant elements within the text. The model exhibits proficiency in identifying prominent entities and sentiments, thereby showcasing its effectiveness in understanding textual information.

Attention Visualization Examples	Target	Prediction
من أسباب تمكين المرأة المزعوم أن تستفيد الدولة الرأسمالية من الضرايب والأجور الي بتدفعها المرأه One of the alleged reasons for empowering women is that capitalist states benefit from the taxes and wages paid by women	Women Empowerment	Against
مابي اخذه ليش الاجبار ذا. اما تطعيم او لاتجي عملك؟؟ هذا مو منطق. انا راضي اموت بكورونا لكن ماموت من الخوف بعدين خلاص اللي اخذ بالتوفيق لك مو لازم كلنا نسوي مثلك I don't want to take it, why this compulsion? Either vaccination or no work?? This isn't logical. I'm willing to die from covid, but I won't die out of fear. Anyway, good luck to those who take it, we don't all have to do the same	COVID-19 Vaccine	Against
التحول الإلكتروني فيه توفير قيمة لكل شي من إنسان او وقود او زحمة طرق او عاجز والفوائد لاتعد ولاتحصى Digital transformation brings value in everything, be it for humans, fuel , traffic congestion, or the disabled, and the benefits are countless and immeasurable.	Digital Transformation	Favor

Table 9 Visualization of attention scores from SMTL-HW model on testing examples of Mawqif dataset, along with their target and correct predictions. Darker colored words are more significant to the model's prediction.

4.3.5 Comparisons with Previous Studies

In order to provide a comprehensive evaluation, we further compared the performance of the best-performing model, i.e., the multi-target SMTL-HW, with the results reported in previous studies. By doing so, we can gain insights into the advancements achieved by our proposed approach compared to existing research. The comparisons are presented in Table 10, where the results are retrieved from the original papers.

Regarding the MAWQIF dataset, no prior systems have been developed for this dataset since it was recently released [71]. Nonetheless, we assessed our top-performing model by comparing it to the best model proposed in the dataset paper [71]. This model is a single-task model that fine-tunes the AraBERT-twitter model using hidden representations encoded from both tweets and targets. It is worth noting that this model follows the same approach as our backbone model, making it a suitable point of comparison.

Furthermore, we evaluated our best performing model on the SemEval-16 dataset by comparing it with previous top-performing models. SemEval, released in 2016, has been extensively utilized in prior studies, enabling meaningful comparisons with other existing approaches. By assessing our model on the SemEval-16 dataset, we can effectively benchmark it against other SOTA models in the field. We compare our model with the following models:

BERT [51]: is a single-task model that extends the pre-trained BERT language model by adding a linear classification layer to the hidden representation of the special [CLS] token.

RoBERTa [77]: is a single-task model that extends the pre-trained RoBERTa language model by adding a linear classification layer to the hidden representation of the special [CLS] token.

JOINT [10]: is a joint model that leverages sentiment information to enhance stance detection without relying on an attention mechanism.

MTIN [78] is a Multi-Task Interaction Network model that simultaneously learns stance and sentiment with a word-level task interaction and task-related graphs.

AT-JSS-Lex [11]: is a multi-task model that incorporates stance and sentiment lexicon to guide its attention mechanism.

MT-LRM-BERT [51]: is a multi-task model that employs a label relation matrix, sentiment classification, and opinion-towards classification to enhance stance detection, while leveraging BERT for network initialization.

As demonstrated in Table 10, the SMTL-HW model achieves the highest F_{Mac} score in stance detection across two datasets. Specifically, on the Mawqif dataset, SMTL-HW exhibits a remarkable 7.3% improvement in F_{Mic} and 5.5% in F_{Mac} compared to the single-task AraBERT-twitter model [71]. Similarly, on the SemEval-16 dataset, the single-task models (i.e., BERT and

Table 10 Comparison with other stance detection models on two benchmark datasets.

Dataset	Category	Model	F_{Mic}	F_{Mac}
Mawqif	Single-task	AraBERT-twitter [71]	79.78	78.89
	Multi-task	**SMTL-HW (ours)**	**85.31**	**86.23**
SemEval-16	Single-task	BERT [51]	71.32	59.59
		RoBERTa [77]	70.01	59.22
	Multi-task	JOINT [10]	69.22	60.16
		MTIN [78]	70.30	64.90
		AT-JSS-Lex [11]	72.33	65.33
		MT-LRM-BERT [51]	**75.10**	67.46
		SMTL-HW (ours)	72.46	**73.23**

RoBERTa) exhibit subpar performance due to their disregard for the significance of sentiment information. Notably, SMTL-HW shows improvements of 12.9% and 13.2% in F_{Mac} compared to the BERT [51] and RoBERTa [77] models, respectively. Although existing multi-task models take into account sentiment information, they still obtain a lower performance on SemEval-16. In terms of F_{Mac}, the SMTL-HW model surpasses JOINT [10], MTIN [78], AT-JSS-Lex [11], and MT-LRM-BERT [51] models by 12.3%, 7.6%, 7.1%, and 5%, respectively. These results highlight the effectiveness of the main components incorporated in our SMTL-HW model, namely the sequential architecture, and task weighting.

5 Conclusion and Future Work

This paper contributes to the field of stance detection by introducing novel multi-task learning models, namely PMTL and SMTL, which incorporate sentiment analysis and sarcasm detection tasks. Additionally, we propose and demonstrate the effectiveness of different task weighting schemes in our models. Notably, our work represents the first attempt to leverage sarcasm detection to enhance the stance detection task and the first to propose task weighting in a multi-task stance detection model. We find that the performance difference between the two architectures, PMTL and SMTL, is relatively small, necessitating further investigation to establish its statistical significance. Moreover, we highlight the potential benefits of incorporating auxiliary tasks to address data scarcity, although its effectiveness may vary depending on the specific dataset and task at hand. Through extensive experiments, we validate the advantages of the proposed models, particularly the SMTL-HW model, which achieves state-of-the-art performance. Empirical results highlight the effectiveness of the main components in our SMTL-HW model, including the sequential architecture and hierarchal task weighting. These features enable obtaining more comprehensive task representations, leading to improved stance detection performance.

Consequently, this research opens up avenues for future exploration in the field of stance detection. Further investigations should be conducted to determine the statistical significance of the performance difference between PMTL and SMTL models, enabling a deeper understanding of their comparative strengths and weaknesses. Additionally, future studies could focus on expanding the scope of multi-task learning by incorporating additional auxiliary tasks across various datasets and tasks, aiming to establish guidelines for their optimal usage. Lastly, integrating other contextual information, such as user profiles, temporal dynamics, or domain-specific knowledge, could be explored to enhance the overall stance detection framework. By incorporating these aspects, the models could better capture the nuanced nature of the stance and improve the overall understanding of textual content.

Acknowledgments The authors would like to acknowledge the support received from the Saudi Data and AI Authority (SDAIA) and King Fahd University of Petroleum and Minerals (KFUPM) under the SDAIA-KFUPM Joint Research Center for Artificial Intelligence Grant JRC-AI-RFP-05.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] AlDayel, A., Magdy, W.: Stance detection on social media: state of the art and trends. Information Processing and Management **58** (2021). https://doi.org/10.1016/j.ipm.2021.102597

[2] Alturayeif, N., Luqman, H., Ahmed, M.: A systematic review of machine learning techniques for stance detection and its applications. Neural Computing and Applications **35**, 5113–5144 (2023). https://doi.org/10.1007/s00521-023-08285-7

[3] Küçük, D., Fazli, C.A.N.: Stance detection: A survey. ACM Computing Surveys **53** (2020). https://doi.org/10.1145/3369026

[4] Wang, R., Zhou, D., Jiang, M., Si, J., Yang, Y.: A survey on opinion mining: From stance to product aspect. IEEE Access **7**, 41101–41124 (2019). https://doi.org/10.1109/ACCESS.2019.2906754

[5] Cortis, K., Davis, B.: Over a decade of social opinion mining: a systematic review. Artificial Intelligence Review **54**, 4873–4965 (2021). https://doi.org/10.1007/s10462-021-10030-2

[6] Ruder, S.: An overview of multi-task learning in deep neural networks (2017)

[7] Niehues, J., Cho, E.: Exploiting linguistic resources for neural machine translation using multi-task learning. (2017). https://doi.org/10.18653/v1/w17-4708

[8] Liu, X., He, P., Chen, W., Gao, J.: Multi-task deep neural networks for natural language understanding, pp. 4487–4496 (2019)

[9] Ebrahimi, J., Dou, D., Lowd, D.: A joint sentiment-target-stance model for stance classification in tweets, pp. 2656–2665 (2016)

[10] Sun, Q., Wang, Z., Li, S., Zhu, Q., Zhou, G.: Stance detection via sentiment information and neural network model. Frontiers of Computer Science **13**, 127–138 (2019). https://doi.org/10.1007/s11704-018-7150-9

[11] Li, Y., Caragea, C.: Multi-task stance detection with sentiment and stance lexicons, pp. 6299–6305 (2019)

[12] Hosseinia, M., Dragut, E., Mukherjee, A.: Stance prediction for contemporary issues: Data and experiments. In: Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media (2020). https://doi.org/10.18653/v1/P17

[13] Aldayel, A., Magdy, W.: Assessing sentiment of the expressed stance on social media, pp. 277–286 (2019). https://doi.org/10.1007/978-3-030-34971-4_19

[14] Sobhani, P., Mohammad, S.M., Kiritchenko, S.: Detecting stance in tweets and analyzing its interaction with sentiment, pp. 159–169 (2016)

[15] Mohammad, S.M., Sobhani, P., Kiritchenko, S.: Stance and sentiment in tweets. ACM Transactions on Internet Technology (TOIT) **17**, 1–23 (2017)

[16] Ghosh, S., Singhania, P., Singh, S., Rudra, K., Ghosh, S.: Stance detection in web and social media: A comparative study, pp. 75–87 (2019). https://doi.org/10.1007/978-3-030-28577-7_4

[17] Li, Y., Tian, X., Liu, T., Tao, D.: Multi-task model and feature joint learning, vol. 2015, pp. 3643–3649 (2015)

[18] Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE Transactions on Knowledge and Data Engineering, 1–20 (2021)

[19] Mahabadi, R.K., Ruder, S., Dehghani, M., Henderson, J.: Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks, pp. 565–576 (2021)

[20] Hacohen-Kerner, Y., Ido, Z., Ya'akobov, R.: Stance classification of tweets using skip char ngrams, pp. 266–278 (2017)

[21] Lai, M., Cignarella, A.T., Irazú, D., Farías, H.: itacos at ibereval2017: Detecting stance in catalan and spanish tweets, pp. 185–192 (2017)

[22] Dey, K., Shrivastava, R., Kaushik, S.: Twitter stance detection-a subjectivity and sentiment polarity inspired two-phase approach, pp. 365–372 (2017). http://www.noslang.com/dictionary

[23] Aldayel, A., Magdy, W.: Your stance is exposed! analysing possible factors forstance detection on social media. Proceedings of the ACM on Human-Computer Interaction **3**, 1–20 (2019)

[24] Bahuleyan, H., Vechtomova, O.: Uwaterloo at semeval-2017 task 8: Detecting stance towards rumours with topic independent features, pp. 461–464 (2017)

[25] Al-Ghadir, A.I., Azmi, A.M., Hussain, A.: A novel approach to stance detection in social media tweets by fusing ranked lists and sentiments. Information Fusion **67**, 29–40 (2021). https://doi.org/10.1016/j.inffus.2020.10.003

[26] Poddar, L., Hsu, W., Lee, M.L., Subramaniyam, S.: Predicting stances in twitter conversations for detecting veracity of rumors: A neural approach, vol. 2018-November, pp. 65–72. IEEE Computer Society, ??? (2018). https://doi.org/10.1109/ICTAI.2018.00021

[27] Borges, L., Martins, B., Calado, P.: Combining similarity features and deep representation learning for stance detection in the context of checking fake news. Journal of Data and Information Quality (JDIQ) **11**, 1–26 (2019). https://doi.org/10.1145/3287763

[28] Sun, L., Li, X., Zhang, B., Ye, Y., Xu, B.: Learning stance classification with recurrent neural capsule network, pp. 277–289 (2019)

[29] Lai, M., Cignarella, A.T., Farías, D.I.H., Bosco, C., Patti, V., Rosso, P.: Multilingual stance detection in social media political debates. Computer Speech and Language **63**, 1–27 (2020). https://doi.org/10.1016/j.csl.2020.101075

[30] Zubiaga, A., Kochkina, E., Liakata, M., Procter, R., Lukasik, M., Bontcheva, K., Cohn, T., Augenstein, I.: Discourse-aware rumour stance

classification in social media using sequential classifiers. Information Processing and Management **54**, 273–290 (2018). https://doi.org/10.1016/j.ipm.2017.11.009

[31] Siddiqua, U.A., Chy, A.N., Aono, M.: Tweet stance detection using multi-kernel convolution and attentive lstm variants. IEICE Transactions on Information and Systems **102**, 2493–2503 (2019). https://doi.org/10.1587/transinf.2019EDP7080

[32] Bhatt, G., Sharma, A., Sharma, S., Nagpal, A., Raman, B., Mittal, A.: Combining neural, statistical and external features for fake news stance identification, pp. 1353–1357 (2018). https://doi.org/10.1145/3184558.3191577

[33] Zhu, L., He, Y., Zhou, D.: Neural opinion dynamics model for the prediction of user-level stance dynamics. Information Processing and Management **57**, 1–13 (2020). https://doi.org/10.1016/j.ipm.2019.03.010

[34] Li, W., Xu, Y., Wang, G.: Stance detection of microblog text based on two-channel cnn-gru fusion network. IEEE Access **7**, 145944–145952 (2019). https://doi.org/10.1109/ACCESS.2019.2944136

[35] Ahmed, M., Chy, A.N., Chowdhury, N.K.: Incorporating hand-crafted features in a neural network model for stance detection on microblog, pp. 57–64 (2020). https://doi.org/10.1145/3442555.3442565

[36] Alkhalifa, R., Kochkina, E., Zubiaga, A.: Opinions are made to be changed: Temporally adaptive stance classification, pp. 27–32 (2021). https://doi.org/10.1145/3472720.3483620

[37] Roy, A., Fafalios, P., Ekbal, A., Zhu, X., Dietze, S.: Exploiting stance

hierarchies for cost-sensitive stance detection of web documents. Journal of Intelligent Information Systems, 1–19 (2021). https://doi.org/10.1007/s10844-021-00642-z

[38] Alec, R., Jeffrey, W., Rewon, C., David, L., Dario, A., Ilya, S.: Language models are unsupervised multitask learners. OpenAI Blog **1** (2019)

[39] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding, vol. 1 (2019)

[40] Clark, K., Luong, M.T., Le, Q.V., Manning, C.D.: Electra: Pre-training Text Encoders as Discriminators Rather than Generators

[41] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer (2019)

[42] Ruder, S., Peters, M., Swayamdipta, S., Wolf, T.: Transfer learning in natural language processing tutorial. (2019)

[43] Hardalov, M., Arora, A., Nakov, P., Augenstein, I.: Cross-domain label-adaptive stance detection, pp. 9011–9028 (2021)

[44] Sun, Q., Xi, X., Sun, J., Wang, Z., Xu, H.: Stance detection with a multi-target adversarial attention network. ACM Transactions on Asian and Low-Resource Language Information Processing (2022). https://doi.org/10.1145/3544490

[45] Liu, Y., Zhang, X.F., Wegsman, D., Beauchamp, N., Wang, L.: Politics: Pretraining with same-story article comparison for ideology prediction and stance detection, pp. 1354–1374 (2022).

http://arxiv.org/abs/2205.00619

[46] Mohtarami, M., Glass, J., Nakov, P.: Contrastive language adaptation for cross-lingual stance detection, pp. 4442–4452 (2019). http://arxiv.org/abs/1910.02076

[47] Vamvas, J., Sennrich, R.: X-stance: A multilingual multi-target dataset for stance detection. (2020). http://arxiv.org/abs/2003.08385

[48] Fang, W., Nadeem, M., Mohtarami, M., Glass, J.: Neural multi-task learning for stance prediction, pp. 13–19 (2019). https://data.quora.com/

[49] Hanselowski, A., P.V.S., A., Schiller, B., Caspelherr, F., Chaudhuri, D.., Meyer, C.M., Gurevych, I.: A retrospective analysis of the fake news challenge stance-detection task. (2018)

[50] Chauhan, D.S., Kumar, R., Ekbal, A.: Attention based shared representation for multi-task stance detection and sentiment analysis, vol. 1143 CCIS, pp. 661–669. Springer, ??? (2019). https://doi.org/10.1007/978-3-030-36802-9_70

[51] Fu, Y., Li, X., Li, Y., Wang, S., Li, D., Liao, J., Zheng, J.: Incorporate opinion-towards for stance detection. Knowledge-Based Systems **246**, 1–11 (2022). https://doi.org/10.1016/j.knosys.2022.108657

[52] Ma, J., Gao, W., Wong, K.F.: Detect rumor and stance jointly by neural multi-task learning, pp. 585–593 (2018). https://doi.org/10.1145/3184558.3188729

[53] Wei, P., Xu, N., Mao, W.: Modeling conversation structure and temporal dynamics for jointly predicting rumor stance and veracity, pp. 4787–4798

(2019). http://arxiv.org/abs/1909.08211

[54] Islam, M.R., Muthiah, S., Ramakrishnan, N.: Rumorsleuth: Joint detection of rumor veracity and user stance, pp. 131–136 (2019). https://doi.org/10.1145/3341161.3342916

[55] Lukasik, M., Bontcheva, K., Cohn, T., Zubiaga, A., Liakata, M., Procter, R.: Gaussian processes for rumour stance classification in social media. ACM Transactions on Information Systems **37**, 1–24 (2019). https://doi.org/10.1145/3295823

[56] Khandelwal, A.: Fine-tune longformer for jointly predicting rumor stance and veracity, pp. 10–19 (2021). https://doi.org/10.1145/3430984.3431007

[57] Ye, K., Piao, Y., Zhao, K., Cui, X.: Graph enhanced bert for stance-aware rumor verification on social media, vol. 12895 LNCS, pp. 422–435. Springer, ??? (2021). https://doi.org/10.1007/978-3-030-86383-8_34

[58] Zhang, H., Qian, S., Fang, Q., Xu, C.: Multi-modal meta multi-task learning for social media rumor detection. IEEE Transactions on Multimedia, 1–11 (2021). https://doi.org/10.1109/TMM.2021.3065498

[59] Wei, P., Lin, J., Mao, W.: Multi-target stance detection via a dynamic memory-augmented network, pp. 1229–1232 (2018). https://doi.org/10.1145/3209978.3210145

[60] Schiller, B., Daxenberger, J., Gurevych, I.: Stance detection benchmark: How robust is your stance detection? KI-Künstliche Intelligenz **35**, 329–341 (2021). https://doi.org/10.1007/s13218-021-00714-w

[61] Sobhani, P., Inkpen, D., Zhu, X.: Exploring deep neural networks for multitarget stance detection. Computational Intelligence **35**, 82–97 (2019). https://doi.org/10.1111/coin.12189

[62] Sobhani, P., Inkpen, D., Zhu, X.: A dataset for multi-target stance detection, vol. 2, pp. 551–557 (2017). https://doi.org/10.18653/v1/e17-2088

[63] Song, W., Song, Z., Liu, L., Fu, R.: Hierarchical Multi-task Learning for Organization Evaluation of Argumentative Student Essays

[64] Yang, M., Chen, L., Chen, X., Wu, Q., Zhou, W., Shen, Y.: Knowledge-enhanced Hierarchical Attention for Community Question Answering with Multi-task and Adaptive Learning

[65] Mao, Y., Wang, Z., Liu, W., Lin, X., Xie, P.: Metaweighting: Learning to weight tasks in multi-task learning, pp. 3436–3448 (2022)

[66] Mao, Y., Wang, Z., Liu, W., Lin, X., Hu, W.: BanditMTL: Bandit-based Multi-task Learning for Text Classification

[67] Antoun, W., Baly, F., Hajj, H.: Arabert: Transformer-based model for arabic language understanding. (2020)

[68] Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)

[69] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R.: Overcoming

catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences of the United States of America **114** (2017). https://doi.org/10.1073/pnas.1611835114

[70] Kendall, A., Gal, Y., Cipolla, R.: Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics

[71] Alturayeif, N.S., Luqman, H.A., Ahmed, M.A.K.: Mawqif: A multi-label Arabic dataset for target-specific stance detection. In: Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pp. 174–184. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (2022). https://aclanthology.org/2022.wanlp-1.16

[72] Mohammad, S.M., Kiritchenko, S., Sobhani, P., Zhu, X., Cherry, C.: Semeval-2016 task 6: Detecting stance in tweets, pp. 31–41 (2016). https://doi.org/10.18653/v1/s16-1003

[73] 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. arXiv preprint arXiv:1907.11692 (2019)

[74] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

[75] Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. (2015)

[76] Ribeiro, M.T., Singh, S., Guestrin, C.: ”why should I trust you?”: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data

Mining, San Francisco, CA, USA, August 13-17, 2016, pp. 1135–1144 (2016)

[77] Chen, P., Ye, K., Cui, X.: Integrating n-gram features into pre-trained model: A novel ensemble model for multi-target stance detection, pp. 269–279. Springer, Deutschland GmbH (2021). https://doi.org/10.1007/978-3-030-86365-4_22

[78] Chai, H., Tang, S., Cui, J., Ding, Y., Fang, B., Liao, Q.: Improving multi-task stance detection with multi-task interaction network, pp. 2990–3000 (2022)