PAPER

Tweet Stance Detection Using Multi-Kernel Convolution and Attentive LSTM Variants

Umme Aymun SIDDIQUA[†a], **Abu Nowshed CHY**[†b], ***Nonmembers*, *and* Masaki AONO**[†c], ***Member***

SUMMARY Stance detection in twitter aims at mining user stances expressed in a tweet towards a single or multiple target entities. Detecting and analyzing user stances from massive opinion-oriented twitter posts provide enormous opportunities to journalists, governments, companies, and other organizations. Most of the prior studies have explored the traditional deep learning models, e.g., long short-term memory (LSTM) and gated recurrent unit (GRU) for detecting stance in tweets. However, compared to these traditional approaches, recently proposed densely connected bidirectional LSTM and nested LSTMs architectures effectively address the vanishing-gradient and overfitting problems as well as dealing with long-term dependencies. In this paper, we propose a neural network model that adopts the strengths of these two LSTM variants to learn better long-term dependencies, where each module coupled with an attention mechanism that amplifies the contribution of important elements in the final representation. We also employ a multi-kernel convolution on top of them to extract the higher-level tweet representations. Results of extensive experiments on single and multi-target benchmark stance detection datasets show that our proposed method achieves substantial improvement over the current state-of-the-art deep learning based methods.

key words: *stance detection, deep learning, neural network model, attention mechanism, nested LSTMs (NLSTMs), densely connected bidirectional LSTM (Bi-LSTM), multi-kernel convolution*

1. Introduction

In recent times, it has become popular for people to express and exchange their feelings, views, and opinions through microblog websites. Among several microblog sites, Twitter is now the most popular, where millions of users post tweets daily, covering trendy topics. That is why; people are increasingly seeking ways to analyze these massive opinion-oriented tweets as a source of information to explain many aspects of human experience on a variety of fundamental social issues.

Tweet stance detection is the task of automatically determining the stance of a tweet whether the content of the tweet is in favor of, against, or none towards a proposition or target [1]. Analyzing user stances from twitter posts is beneficial to address several real-life scenarios including political, economic, and social events [2]. We can consider stance detection as a sub-domain of sentiment analysis, but there is a significant difference between them. In sentiment analysis, the major goal is to classify the polarity of a tweet sentiment either positive, negative, or neutral based on its contents, whereas the identification of stance is dependent on the specific target. The target may be a person, an organization, a government policy, and so on. But it is not necessary that the target explicitly mentioned in the tweet. Moreover, a person can express the same stance towards a target by using positive and negative sentiment bearing words. To illustrate the above scenario, we articulate sample tweets for the target "Donald Trump" in Table 1. For example, Tweet 2 implies a stance favor to Donald Trump through opposing Hillary Clinton. We see that the sentiment polarity of the Tweet 2 is negative and the target Donald Trump is not mentioned in the tweet.

Twitter stance detection poses unique challenges to the research community since tweets are short and informal user-generated text, which usually tend not to follow the grammatical rules. Moreover, tweets contain plenty of idiosyncratic abbreviations as well as other twitter specific syntaxes such as #hashtags and emoticons. To address the challenges of stance detection in twitter, Mohammad et al. [1] presented a tweet stance detection task in SemEval-2016 that focused on a single target. Top performing systems in this task proposed several deep learning based approaches by using convolutional neural networks (CNN) [3], recurrent neural networks (RNN) [4], and other traditional models.

Besides the single target tweet stance detection, there are cases that a tweeter (tweet author) may express his stance towards different targets in the same tweet within a domain where several targets are closely related to each other such as candidates in a general election. For example, in Fig. 1 we see that the same tweet poses different stances towards two different targets. Here, the stance of the tweet is in favor of Donald Trump while against towards the Hillary Clinton. By considering such scenarios, Sobhani et al. [5] introduced a multi-target stance detection (MTSD) task to address the plausible dependency of related targets.

However, most of the related work of tweet stance detection explored the traditional deep learning models in their methods, which motivate us to address the tweet stance detection problem using state-of-the-art techniques.

The main contribution of this paper is that we propose a neural network model that combines the attention based densely connected Bi-LSTM [6] and nested LSTMs [7] models with the multi-kernel convolution in a unified

Manuscript received March 21, 2019.
Manuscript revised July 7, 2019.
Manuscript publicized September 25, 2019.
[†]The authors are with Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.
a) E-mail: aymun@kde.cs.tut.ac.jp
b) E-mail: nowshed@kde.cs.tut.ac.jp
c) E-mail: aono@tut.jp
DOI: 10.1587/transinf.2019EDP7080

Table 1 Example tweets for the target "Donald Trump" to illustrate the difference between stance detection and sentiment analysis.

Sentiment	Example Tweets	Stance
Positive	Tweet 1: Trump wasn't precise but he's no bigot. employs thousands. contributed millions to charity…	Favor
Negative	Tweet 2: No more Bush or Hillary Clinton #MakeAmericaGreatAgain	Favor
Negative	Tweet 3: @realDonaldTrump is such an idiot. He's risking way too much for something he'll never get.	Against
Positive	Tweet 4: There is still more to be done to make this country strong, powerful and great again. #TrumpCarson2016	Against

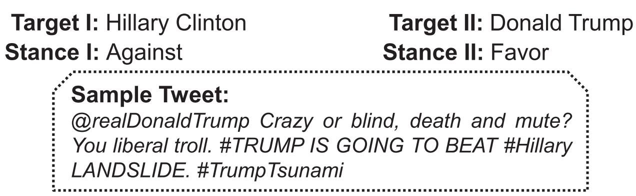


Fig. 1 Example of target-specific tweet stance detection.

architecture. Experimental results on both the single and multi-target benchmark stance detection datasets demonstrate the efficacy of our method over the state-of-the-art methods.

The rest of the paper is structured as follows: **Sect. 2** gives an overview of the related work. In **Sect. 3**, we introduce our proposed stance detection framework. **Section 4** includes experiments and evaluation as well as the comparisons with the state-of-the-arts to show the effectiveness of our proposed method. Some concluded remarks and future directions of our work are described in **Sect. 5**.

2. Related Work

Stance detection and argumentation mining have emerged recently as one of the most prominent research task in opinion mining. There has been a growing interest in these fields, as they can be advantageous particularly for decision making in various real-life scenarios. Previously, stance detection has been studied extensively in a variety of problem domain including congressional debates [8], online forums debates [9]–[13], online news articles [14], online news comments [15], and document-level student essays [16]. To address these problems, researchers have utilized the lexicon and sentiment expression [9], have constructed the rule-based classifier [10], have exploited dialogue relations [11], and have used topic modeling [15].

More recently, twitter has been considered as a corpus of studying public opinion, as increasingly many people express their opinions implicitly or explicitly in their tweets. But twitter stance detection is less explored problem as compared to the stance detection on other problem domains. In one of a few works in stance detection in twitter, Rajadesingan et al. [17] determined user-level stance based on the assumption that if several users retweet about a controversial topic, they are on the same side. Djemili et al. [18] used a set of linguistic rules to identify the ideological stance of tweets. However, none of these works attempts to determine the stance from a single tweet.

Mohammad et al. [1] presented a twitter stance detection task in the SemEval-2016, where given a tweet and a target entity, a system needs to determine whether the tweet author is in favor of or against towards the target. Several models presented in this competition. Some participants proposed deep learning based approaches including convolutional neural networks (CNN) [3], recurrent neural networks (RNN) [4], and so on. Other participants addressed the problem by utilizing the classical supervised classifiers including SVM, Naive-Bayes, and others along with the ensemble of features [1].

Later, Du et al. [19] utilized the target-augmented embeddings in an attention based neural network, whereas Zhou et al. [20] proposed an attention mechanism at the semantic level in the bidirectional GRU-CNN structure to perform target-specific stance detection on tweets. More recently, Dey et al. [2] proposed a two-phase LSTM based model with attention. In the first phase, they performed the subjectivity analysis and in the second phase, they detected the stance of the tweets. Along with this direction, Wei et al. [21] proposed an end-to-end neural memory model via target and tweet interactions. Besides the deep learning based approach, Siddiqua et al. [22] leveraged the syntactic tree representation of tweets to detect the tweet stance where a parts-of-speech (POS) generalization technique along with hashtag segmentation employed for effective tree representation.

By considering the dependency of related targets, Sobhani et al. [5] introduced a multi-target stance detection (MTSD) task and proposed an attentive encoder-decoder network to capture the dependencies among stance labels regarding multiple targets. Later, Wei et al. [23] proposed a dynamic memory-augmented network that utilized a shared external memory to capture and store multi-targets stance indicative clues dynamically.

However, the majority of the existing related approaches attempts to utilize the traditional neural network models to detect the stance of a tweet, whereas recently introduced deep learning models such as densely connected Bi-LSTM [6] and nested LSTMs [7] achieved significant improvements to address the vanishing-gradient and overfitting problems as well as dealing with long-term dependencies effectively. Therefore, to bridge this research gap, in this paper, we propose a neural network model that adopts these LSTM variants with attention mechanism and multi-kernel convolution in a unified architecture.

3. Our Proposed Stance Detection Framework

In this section, we describe the details of our proposed neural network model. The goal of our proposed approach is to determine the stance of a tweet whether it is in favor of, against, or none (neutral) towards a single or multiple targets. Figure 2 depicts an overview of our proposed framework.

At first, we utilize the multi-kernel convolution filters with four different kernel sizes including [2,3,4,5] to extract the higher-level feature sequences from the target appended tweet embeddings. The generated feature sequences are then concatenated and fed into the densely connected Bi-LSTM and nested LSTMs to learn long-term dependencies. In order to focus on the important portions of the last hidden states sequences, we employ a feed-forward attention mechanism that aggregates the hidden states according to their relative importance weight. Final representations of these two modules are concatenated and send to the fully-connected stance prediction module to determine the final stance label. For the simplicity of discussion, we named our proposed neural network architecture as MKC-LSTMVs-ATT. In the following sections, we describe each component elaborately.

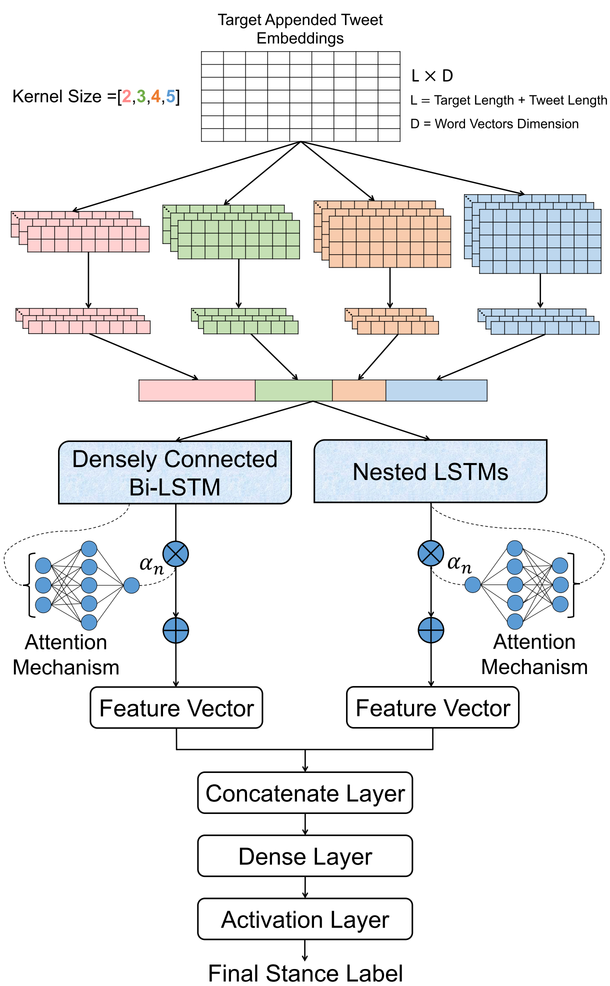


Fig. 2 Proposed stance detection framework.

3.1 Embedding Layer

Distributed representation of words known as word embedding is treated as one of the most popular representations of documents vocabulary due to its capability of capturing the context of a word within a document as well as estimating the semantic similarity and relation with other words. Such a vector representation of documents can help learning algorithms to achieve better performances in various natural language processing (NLP) applications [24]–[26].

In our proposed framework, we utilize a pre-trained word embedding model to obtain the high-quality distributed vector representations of tweets. Moreover, prior works [19], [21] already established the significance of target information for stance detection. To integrate the target information, we generate a unified word vector matrix by concatenating the vector representations of the target and tweet. The dimensionality of the matrix will be $L \times D$, where length L is the sum of the target length and tweet length, and D denotes the word-vectors dimension. Here, target length means the number of words available in a target and tweet length means the number of words available in a tweet. For example, we can consider a sample target "Donald Trump" which length is 2 and a sample tweet "Make America Great Again #Trump #USA" which length is 6, therefore, length $L = 2 + 6 = 8$.

3.2 Multi-Kernel Convolution

The convolution layers usually applied to extract the higher level features from the given input matrix. Since kernel sizes, i.e., the size of the convolution filters have a significant effect on performance, we apply filters with different sizes to get the different kinds of effective features. Previous studies already demonstrated the effectiveness of multi-kernel convolution over the single one [27]–[29].

In our multi-kernel convolution, we adopt the idea proposed by Kim et al. [27], where the input is the target appended tweet embedding matrix generated in the embedding layer. We then perform the convolution on it by using a filter. We apply multiple convolutions based on four different kernel sizes: 2, 3, 4, and 5. After performing convolutions, each filter generates the corresponding feature maps and a max pooling function is then applied to form a univariate feature vector. Finally, the feature vectors generated from each kernel are concatenated to form a single high-level feature vector.

3.3 Densely Connected Bi-LSTM

With the emerging trend of deep learning, LSTM based models are the most popular for sequential tasks. To alleviate the vanishing-gradient and overfitting problems effectively, recently densely connected structure of LSTM models gets attention among the researchers [30]–[32]. Such structures enable the effective connection through concatenation operation from lower to upper layers features without any loss of information on lower-layer features, which make it effective to learn better long-term dependencies.

In our proposed architecture, we utilize the densely connected Bi-LSTM [6] (DC-Bi-LSTM) model. A DC-Bi-LSTM model consists of multiple Bi-LSTM layers, where the representation of each layer is estimated by concatenating its hidden states and all the preceding layers' hidden states. Hence, for the first Bi-LSTM layer, the input is a higher-level feature sequences generated from multi-kernel convolution (MKC) and the output is $\{h^1 = h^1_1, h^1_2, \ldots, h^1_s\}$. For the second Bi-LSTM layer, the input is the concatenation of higher-level feature sequences from MKC and the output from first Bi-LSTM layer to generate the corresponding output. Rest of the layers are processed accordingly. We can define the above process as follows:

$$\begin{aligned}
&h^l_t = [\overrightarrow{h^l_t}; \overleftarrow{h^l_t}],\ h^0_t = \text{MKC feature sequence},\\
&\overrightarrow{h^l_t} = \text{lstm}(\overrightarrow{h^l_{t-1}}, M^{l-1}_t),\\
&\overleftarrow{h^l_t} = \text{lstm}(\overleftarrow{h^l_{t+1}}, M^{l-1}_t),\\
&M^{l-1}_t = [h^0_t; h^1_t; \ldots; h^{l-1}_t].
\end{aligned}$$

Therefore, from an L layer DC-Bi-LSTM model, the output is $\{h^L = h^L_1, h^L_2, \ldots, h^L_s\}$.

3.4 Nested LSTMs

Besides the densely connected Bi-LSTM (DC-Bi-LSTM), we also utilize the state-of-the-art nested LSTMs (NLSTMs) [7] model that achieved significant improvement over the single-layer or stacked LSTM architectures to learn longer-term dependencies.

The nested LSTMs (NLSTMs) architecture [7] creates a temporal hierarchy of memories. In NLSTMs, the LSTM memory cells have access to their inner memory, where they can selectively read and write relevant long-term information. In an LSTM, the cell state and the gates are governed by the following equations [33], [34]:

$$\begin{aligned}
&i_t = \sigma_i(x_t W_{xi} + h_{t-1} W_{hi} + b_i)\\
&f_t = \sigma_f(x_t W_{xf} + h_{t-1} W_{hf} + b_f)\\
&g_t = \tanh(x_t W_{xc} + h_{t-1} W_{hc} + b_c)\\
&c_t = f_t \odot c_{t-1} + i_t \odot g_t\\
&o_t = \sigma_o(x_t W_{xo} + h_{t-1} W_{ho} + b_o)\\
&h_t = o_t \odot \tanh(c_t)
\end{aligned}$$

where i_t, f_t, o_t, g_t, c_t, and h_t denote the input gate, forget gate, output gate, cell input activation, the cell state, and the current hidden state, respectively, at the current time step t. The symbol σ_i, σ_f, and σ_o are set to the logistic sigmoid function to set the gating values in $[0, 1]$ and $\odot$ is the element-wise multiplication.

While the value of the outer memory cell in the LSTM is estimated as $c^{outer}_t = f_t \odot c_{t-1} + i_t \odot g_t$, memory cells of the NLSTMs use the concatenation $(f_t \odot c_{t-1}, i_t \odot g_t)$ as input to an inner LSTM (or NLSTM) memory cell, and set $c^{outer}_t = h^{inner}_t$. Therefore, in compared to the LSTM and stacked LSTM, the inner memories of NLSTMs operate on longer time-scales and effectively capture the context information from the input tweet texts.

3.5 Feed-Forward Attention

Recently, the attention mechanism has been introduced in the neural network models for effectively modeling the long-term dependencies by enabling the model to learn what to attend based on the input text [35]–[37]. In order to amplify the contribution of important elements in the final representation of both the DC-Bi-LSTM and NLSTMs module, we employ a feed-forward attention mechanism proposed by Raffel and Ellis [38] to aggregate all the hidden states according to their relative importance weight.

Let us consider h_t the hidden state of either the DC-Bi-LSTM or the NLSTMs module at some time t where $t \in [1, T]$ and T is the number of time steps in the input sequence. The weight α_t corresponding to each h_t is estimated as follows:

$$\alpha_t = \frac{\exp(\psi_t)}{\sum_{t=1}^{T} \exp(\psi_t)}$$

where

$$\psi_t = \tanh(w^T_{attn} \cdot h_t + b_{attn})$$

Here, w_{attn} and b_{attn} are the attention parameters to learn.

Finally, the hidden states are estimated through the weighted average of the sequence h_t to generate an attention based single feature vector c as follows:

$$c = \sum_{t=1}^{T} \alpha_t h_t$$

3.6 Stance Prediction and Model Training

After getting the final tweet representation from both the attention based DC-Bi-LSTM and NLSTMs module, we concatenate them and pass it to a fully connected softmax layer for stance detection. The output of the softmax layer is the probability distribution over all the three stance categories and we consider the category with the maximum value as the final stance label for a given input tweet.

We consider the cross-entropy as the loss function and train the model by minimizing the error, which is defined as follows:

$$E(x^{(i)}, y^{(i)}) = \sum_{j=1}^{k} 1\{y^{(i)} = j\} \log(\tilde{y}^{(i)}_j)$$

where $x^{(i)}$ is the training sample with its true label $y^{(i)}$. $\tilde{y}_j^{(i)}$ is the estimated probability in $[0, 1]$ for each label j. $1\{condition\}$ is an indicator which is 1 if true and 0 otherwise.

We use the stochastic gradient descent (SGD) algorithm to learn the model parameter and adopt the Adam optimizer [39], [40]. We apply the commonly used dropout [41] and L2 weight regularization techniques in our proposed model. Dropout helps to prevent complex co-adaptations while using a small set of training data.

4. Experiments and Evaluation

We evaluate our proposed stance detection framework on two benchmark twitter datasets: single target stance detection dataset used in SemEval-2016 Task 6-A [1] and multi-target stance detection dataset provided by Sobhani et al. [5]. At first, we describe the details of our model's configuration and dataset preprocessing task in the next two subsections. Later, we present the details of each dataset along with extensive experimental results in the subsequent subsections.

4.1 Model Configuration

In the following, we describe the set of parameters that we have used to design our proposed neural network model, MKC-LSTMVs-ATT. Our designed model was based on Tensorflow [42] and trained on a GPU [43] to capture the benefit from the efficiency of parallel computation of tensors. We performed hyper-parameter optimization using a simple grid search. We used the 300-dimensional fastText embedding model pre-trained on Wikipedia with skip-gram [26] to initialize the word embeddings in the embedding layer described in Sect. 3.1. For the multi-kernel convolution described in Sect. 3.2, we employed 4 different kernel sizes including (2,3,4,5), and the number of filters was set to 600. In our model, DC-Bi-LSTM module contains 5 Bi-LSTM layers and nested LSTMs module contains 2 layers. We trained all models for a maximum of 45 epochs with a batch size of 32 and an initial learning rate 0.001 by Adam optimizer. Both the DC-Bi-LSTM layers and the nested LSTMs layers were dropped out with a probability of 0.2. $L2$ regularization with a factor of 0.01 was applied to the weights in the softmax layer. In this paper, we reported the results based on these settings.

In order to make the comparison, we consider a strong baseline, which is the combination of CNN and LSTM [44] and obtained competitive performances on several classification tasks. We used a similar kind of parameter settings in our baseline according to the recommendation by Zhou et al. [44]. Our baseline system contains one CNN layer and one LSTM layer. The CNN is constructed on top of 300-dimensional word vectors pre-trained on the Google News Dataset[†]. The kernel size was set to 3 and the number of filters was set to 150. Both the CNN layer and the LSTM layer were dropped out with a probability of 0.5. RMSprop optimizer [40] was applied with the stochastic gradient descent (SGD) algorithm to learn the model parameter and $L2$ regularization with a factor of 0.001 was applied to the weights in the softmax layer. Unless otherwise stated, default settings were used for the other parameters.

[†]https://code.google.com/archive/p/word2vec/

4.2 Data Preprocessing

Data preprocessing stage is initiated with tokenization. As tweets are informal user generated contents, people use lots of emoticons and twitter specific syntaxes (@, #hashtag, etc.). But meaningful English words do not contain these characters. We remove these characters from tweets except #hashtag. Since #hashtags contain important information about the tweets subjects, we replace the #hashtag with the segmented words as described in the next paragraph. Moreover, in stance detection, stopwords play a negative role because they do not carry any opinion-oriented information and may damage the performance of the classifiers. For stopword removal, we utilize the NLTK's [45] standard stoplist.

The short length constraint of the tweet makes characters expensive. To overcome this constraint, people usually use #hashtags to express their thoughts concisely. A twitter #hashtag is a type of label or metadata tag used by users within a tweet to highlight the trending events or issues on twitter. Similar kind of hashtags indicates the similar context of the tweet. Segmenting the hashtags might therefore be helpful in better understanding and extraction of information from tweets. For hashtag segmentation, we make use of a hashtag segmentation tool provided by Baziotis et al. [46]. Their method utilized the Viterbi algorithm [47] to estimate the most probable sequence based on the word statistics from a large twitter corpus. By utilizing this tool, we identify and segment the #hashtag into the meaningful set of words and replace the #hashtag with the segmented words.

4.3 Single Target Stance Detection

4.3.1 Dataset Collection

To validate the effectiveness of our proposed method for the single target stance detection, we made use of a widely popular benchmark twitter dataset used in the SemEval-2016 Task 6-A [1]. The training set consists of 2914 tweets and the test set consists of 1249 tweets relevant to 5 targets. Each tweet was annotated as Favor, Against, or None towards the specific target. Detailed statistics about the distribution of tweets in the training and test sets for each target is presented in Table 2.

Following the SemEval-2016 Task 6-A benchmark [1], we employed the macro-average of F1-score (the harmonic mean of precision and recall) for the Favor and Against stance classes as the evaluation measure defined as follows:

Table 2 The statistics of train and test sets for SemEval-2016 Task 6-A twitter stance detection dataset.

Target	#Total	#Train	Train Instances (%)			#Test	Test Instances (%)		
			Favor	Against	None		Favor	Against	None
Atheism	733	513	17.9	59.3	22.8	220	14.5	72.7	12.7
Climate Change is a Real Concern	564	395	53.7	3.8	42.5	169	72.8	6.5	20.7
Feminist Movement	949	664	31.6	49.4	19.0	285	20.4	64.2	15.4
Hillary Clinton	984	689	17.1	57.0	25.8	295	15.3	58.3	26.4
Legalization of Abortion	933	653	18.5	54.4	27.1	280	16.4	67.5	16.1
Total	4163	2914	25.8	47.9	26.3	1249	24.3	57.3	18.4

Table 3 Performance on different experimental settings for single target stance detection. The best results are highlighted in boldface.

Method	F_{Favor}	$F_{Against}$	F_{Avg}
Baseline (CNN+LSTM)	59.36	74.93	67.15
MKC-LSTMVs-ATT	**66.56**	**77.66**	**72.11**
– Target Embedding	64.87	77.36	71.12
– MKC	60.49	73.78	67.13
– ATT	63.94	76.92	70.43
– (NLSTMs+ATT)	61.51	74.43	67.97
– (DC-Bi-LSTM+ATT)	64.47	77.52	71.00

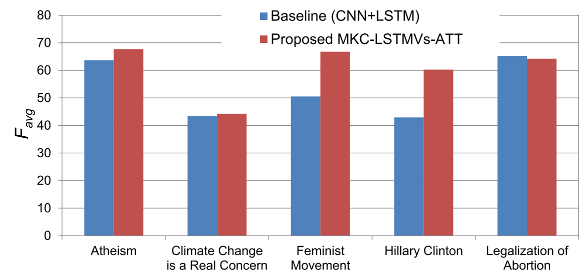


Fig. 3 (Single target) Target-wise performance comparison.

$$F_{Avg} = \frac{F_{Favor} + F_{Against}}{2}$$

In order to ensure appropriate evaluation and comparison, we used the evaluation scripts provided by the SemEval task organizers.

4.3.2 Results and Analysis

We divided the whole dataset across targets and trained the model accordingly. We used 5% of the training samples as the validation set. The summarized experimental results of our proposed neural network model (MKC-LSTMVs-ATT) on single target stance detection are presented in Table 3.

At first, we report the results based on baseline (CNN+LSTM) [44]. Next, we report the results of our proposed MKC-LSTMVs-ATT model. It showed that our proposed method outperformed the baseline by a large margin in terms of evaluation measure.

In order to estimate the effect of each component, we performed the component ablation study on our proposed model. In this regard, we removed one component each time and repeated the experiment. From Table 3, it can be observed that when removing target embedding, multi-kernel convolution (MKC), attention mechanism (ATT), NLSTMs with corresponding attention (NLSTMs+ATT), and DC-Bi-LSTM with corresponding attention (DC-Bi-LSTM+ATT), the results decreased by 0.99%, 4.98%, 1.68%, 4.14%, and 1.11%, respectively. Thus deduced the importance of each of the component in our model.

We also compared the target-wise performance of our proposed method (MKC-LSTMVs-ATT) with the baseline. Figure 3 showed the target-wise comparative results. Our proposed method outperforms the *baseline* by a large margin for the target "Feminist Movement" and "Hillary Clinton." For the target "Atheism," our model outperforms the baseline by a moderate margin. However, our model obtained nearly similar performance for the other two targets. Though our method marginally outperforms the baseline for the target "Climate Change is a Real Concern," but slightly lags in the target "Legalization of Abortion."

4.3.3 Comparison with Related Work

We compared the performance of our proposed method (MKC-LSTMVs-ATT) with the state-of-the-art stance detection methods including SemEval-2016 baselines [1], top 3 performing teams in SemEval-2016 named as MITRE [4], Pkudblab [3], and TakeLab [48], recently proposed deep learning based methods TGMN-CR [21], *n*-grams+embeddings [49], AS-biGRU-CNN [20], T-PAN [2], and TAN [19] as well as other related methods such as EnsembleTKPGHS [22]. The comparative results are presented in Table 4. The results showed that our proposed method achieved substantial improvement over all the related methods in terms of primary evaluation measure F_{avg}. We also reported the overall $F1$ score for the Favor and Against stance classes denoted as F_{Favor} and $F_{Against}$ along with the target-wise $F1$ score.

At first, we compared the performance of our method with the SemEval-2016 baselines [1]. The organizers used four baselines, where the first one is the majority class that assigns the label based on majority vote. In the SVM-unigrams method, they utilized the word unigram features with the target-specific SVM classifier. SVM-ngrams utilize both the word n-grams (1-, 2-, and 3-gram) and character n-grams (2-, 3-, 4-, and 5-gram) features with the target-specific SVM classifier. Finally, in the SVM-ngrams-comb

Table 4 (Single target) Comparative performance of our model against the state-of-the-art methods. The best results are highlighted in boldface.

Method	Overall			Target-wise F1 Score				
	F_{Favor}	$F_{Against}$	F_{Avg}	Atheism	Climate Change.	Feminist Movement	Hillary Clinton	Legal. of Abortion
MKC-LSTMVs-ATT	**66.56**	77.66	**72.11**	67.73	44.27	**66.76**	60.28	64.23
SemEval-2016 Baselines [1]								
Majority class	52.01	**78.44**	65.22	42.11	42.12	39.10	36.83	40.30
SVM-unigrams	54.49	72.13	63.31	53.25	38.39	55.65	57.02	60.09
SVM-ngrams	62.98	74.98	68.98	65.19	42.35	57.46	58.63	66.42
SVM-ngrams-comb	54.11	70.01	62.06	53.27	47.76	52.82	56.50	63.71
Top 3 Performing Teams in SemEval-2016 [1]								
MITRE [4]	59.32	76.33	67.82	61.47	41.63	62.09	57.67	57.28
Pkudblab [3]	61.98	72.67	67.33	63.34	52.69	51.33	64.41	61.09
TakeLab [48]	60.93	72.73	66.83	67.25	41.25	53.01	**67.12**	61.38
Related Deep Learning based Methods								
TGMN-CR [21] (IJCNN,18)	65.52	76.55	71.04	64.60	43.02	59.35	66.21	66.21
n-grams+embeddings [49] (ACM TOIT,17)	-	-	70.30	**68.30**	43.80	58.40	57.80	66.90
AS-biGRU-CNN [20] (WISE,17)	-	-	69.42	66.76	43.40	58.83	57.12	65.45
T-PAN [2] (ECIR,18)	-	-	68.84	61.19	**66.27**	58.45	57.48	60.21
TAN [19] (IJCAI,17)	-	-	68.79	59.33	53.59	55.77	65.38	63.72
Other Related Method								
EnsembleTKPGHS [22] (DMBD,18)	62.11	77.96	70.03	59.53	42.12	56.84	53.64	**71.89**

method, they combined the whole training set among the targets for training instead of the target-specific training. Next, we performed a comparison with the top three performing teams in the SemEval-2016 stance detection task [1]. MITRE [4] achieved top score by utilizing two recurrent neural networks (RNN) classifiers: the first was trained to predict task-relevant hashtags, which was used to initialize the second RNN classifier trained with the provided training data. The 2nd best, Pkudblab [3] used a convolutional neural network (CNN) model with a voting scheme. TakeLab [48] utilized the off-the-shelf machine learning models with the genetic algorithm. However, most of the above methods used the traditional model to address the stance detection task, whereas we combine several state-of-the-art neural network models in an effective way which significantly improve the predictive power of our model. Results in Table 4 showed that our model gained 3.13% and 4.29% improvement while compared with the top SemEval-2016 baseline (SVM-ngrams) and the best performing system MITRE, respectively.

Furthermore, we also compared our proposed method with the recently proposed related deep learning methods. TGMN-CR [21] is an end-to-end neural memory model which learns the target-specific tweet representation and extracts stance-indicative clues through multiple interactions between target and tweet words. In the *n*-grams+embeddings [49] method, Mohammad et al. utilized a linear SVM classifier with the word n-grams features, character n-grams features, and embedding based features drawn from additional unlabeled data. In the AS-biGRU-CNN [20] model, Zhou et al. proposed a semantic level attention mechanism that attended the useful semantic features of informative tokens in a tweet. They used this attention mechanism to inject the target information into the bidirectional GRU-CNN model. T-PAN [2] is a two-phase attention-embedded LSTM-based approach, where the first phase corresponds to subjectivity analysis and the second phase corresponds to stance detection. In the subjectivity analysis phase, they considered the Favor and Against tweets as subjective and None tweets as non-subjective. TAN [19] method utilized a novel target-specific attention extractor to focus on the critical parts of a tweet related to the target and combined it with a bidirectional LSTM model. However, in our proposed model, we encode the target information in the embedding layer and later utilized a feed-forward attention mechanism with the densely connected Bi-LSTM and nested LSTMs. Results showed that our proposed method achieved at least 1.07% (TGMN-CR) and at best 3.44% (TAN) improvement while compared with the related deep learning methods thus validates the effectiveness of our method.

Besides the deep learning based methods, we also compared with the one state-of-the-art related method, EnsembleTKPGHS [22]. It leveraged the syntactic tree representation of tweets, where a parts-of-speech (POS) generalization technique and hashtag segmentation are used for effective tweet representation. Finally, a majority voting count based prediction scheme based on support vector machine (SVM) classifier with three tree kernel functions is employed to identify the tweet stance. Results showed that our method surpassed this method by 2.08% thus demonstrated the efficacy of our method.

Table 5 Multi-target stance detection dataset.

Target Pair	#total	#train	#dev	#test
Clinton-Sanders	1366	957	137	272
Clinton-Trump	1772	1240	177	355
Cruz-Trump	1317	922	132	263
Total	4455	3119	446	890

Table 6 (Multi-target) Performance of our model against the baseline (CNN+LSTM). The best results are highlighted in boldface.

Method	Overall	Target Pair		
		Clinton Sanders	Clinton Trump	Cruz Trump
Baseline (CNN+LSTM)	46.03	43.02	47.35	47.71
MKC-LSTMVs-ATT	**58.72**	**57.74**	**60.05**	**58.36**

4.4 Multi-Target Stance Detection

4.4.1 Dataset Collection

In order to assess our method for the multi-target stance detection, we made use of a benchmark Twitter dataset provided by Sobhani et al. [5], where each tweet is annotated with two stance labels towards two targets of a pair. The tweets in the dataset are related to the 2016 US election. Overall, the dataset contains 4455 tweets for the three target pairs including Hillary Clinton - Bernie Sanders, Hillary Clinton - Donald Trump, and Ted Cruz - Donald Trump. The train, development, and test set contains 3119, 446, and 890 tweets, respectively. Detailed statistics of the dataset shown in Table 5.

As the evaluation measure, we used a similar kind of approach reported in the dataset paper [5]. Following this, the F_{Avg} of each target is estimated according to the SemEval-2016 Task-A benchmark [1]. Then, the F_{Avg} of the two targets of a target-pair is averaged to estimate the average score of a target-pair. Finally, the average scores of all the target-pairs are averaged to estimate the overall score.

4.4.2 Results and Analysis

We divided the whole dataset based on each target-pairs. The dataset is then separated into two parts for each target in the target-pair. We trained and evaluated the model for each target in the target-pair and combined the results to estimate the overall performance. The experimental results of our proposed neural model (MKC-LSTMVs-ATT) on multi-target stance detection dataset are presented in Table 6.

We used the similar kind of baseline for comparison that we used in the single target stance detection. Results showed that our proposed method surpassed the baseline by a large margin in terms of both the overall score and target pair specific score, which demonstrated the effectiveness of our method for the multi-target stance detection.

Table 7 (Multi-target) Comparative performance of our model against the state-of-the-art methods. The best results are highlighted in boldface.

Method	Overall	Target Pair		
		Clinton Sanders	Clinton Trump	Cruz Trump
MKC-LSTMVs-ATT	**58.72**	**57.74**	60.05	**58.36**
Related Deep Learning based Methods				
Seq2Seq [50] (CI,18)	54.81	54.72	56.60	53.12
DMAN [23] (SIGIR,18)	56.73	56.25	**60.30**	53.64

4.4.3 Comparison with Related Work

We compared the performance of our proposed method (MKC-LSTMVs-ATT) with the state-of-the-art multi-target stance detection methods including Seq2Seq method proposed by Sobhani et al. [50] and DMAN method proposed by Wei et al. [23]. The comparative results are presented in Table 7.

Seq2Seq [50] is the attention-based encoder-decoder deep neural network model for multi-target stance detection. The encoder converted the input text into a vector representation, and the decoder generates stance labels towards multiple targets. DMAN [23] utilized the dynamic memory augmented network to capture and store stance-indicative information for related targets. Results showed that our MKC-LSTMVs-ATT method substantially surpassed these methods and gained 3.91% and 1.99% improvement over the Seq2Seq and DMAN methods, respectively.

Moreover, we also reported the comparative performance of our proposed model and related methods in terms of F_1 score of each target and the average over the target pairs. The results are presented in Table 8. It showed that our model outperforms the related methods over several targets and two target pairs thus validated the efficacy of our method.

4.5 Discussion

To further analyze the performance of our proposed MKC-LSTMVs-ATT method, we provide the normalized confusion matrix based on the single and multi-target stance detection dataset in Fig. 4 and Fig. 5, respectively.

Figure 4 shows that 25% of Favor tweets are misclassified, whereas 31% Against tweets are misclassified for single target stance detection. Most of the misclassification occurs in the lower triangle of this matrix, which indicates that our proposed model is slightly biased towards classifying tweets as Favor or Against than the human annotators. Approximately 22% of Against tweets and 27% of None tweets are incorrectly classified as Favor class. To explore more about the reason for misclassification, we articulate some example tweets in Table 9.

From the observation, we see that some tweets are ambiguously classified by the human annotators. For example, it seems to be clear that Tweet 1 favors towards the

Table 8 (Multi-target) Performance of our proposed method and related methods on each target in terms of $F1$ score and the average over the target pairs. The best results are highlighted in boldface.

Method	Clinton-Sanders			Clinton-Trump			Cruz-Trump		
	Clinton	Sanders	Avg	Clinton	Trump	Avg	Cruz	Trump	Avg
MKC-LSTMVs-ATT	**63.60**	51.88	**57.74**	**59.99**	**60.11**	60.05	**58.39**	58.34	**58.36**
Baseline(CNN+LSTM)	46.95	39.09	43.02	49.98	44.73	47.35	46.73	48.70	47.71
Seq2Seq [50] (CI,18)	55.59	**53.86**	54.72	54.46	58.74	56.60	47.02	**59.21**	53.12
DMAN [23] (SIGIR,18)	-	-	56.25	-	-	**60.30**	-	-	53.64

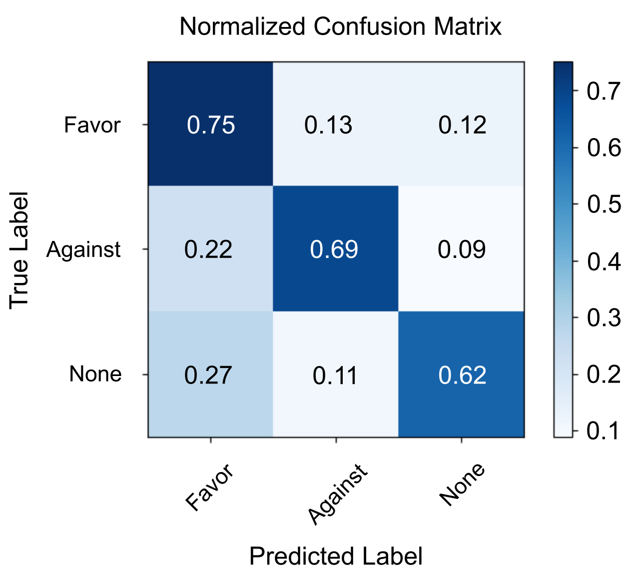


Fig. 4 Normalized confusion matrix for single target stance detection.

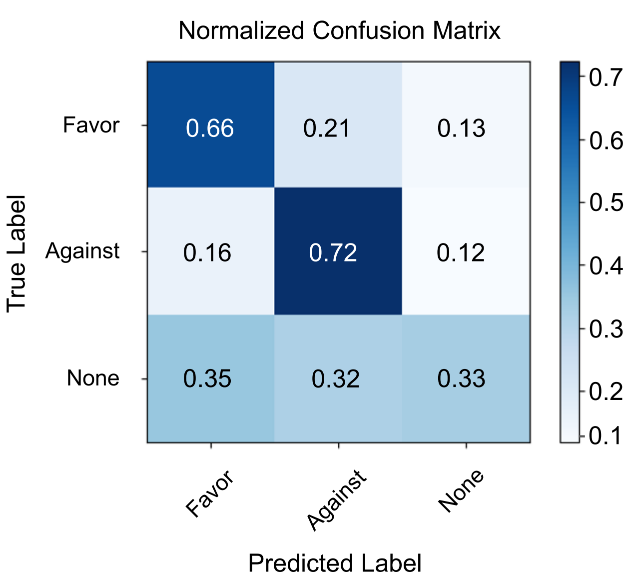


Fig. 5 Normalized confusion matrix for multi-target stance detection.

target "Atheism" and our system also classified it as Favor. However, the ground truth for this tweet is Against. The same thing also happened in Tweet 3, where it is hard to decide that the tweet is Against the target "Hillary Clinton." For the Tweet 2, we see that our system erroneously classified the tweet as Favor. We think that our system focuses highly on the phrase "froze In the summer?!!" than the word "stock market." Tweet 4 is written as a conversational manner, which our system failed to capture thus misclassified it, whereas Tweet 5 is very short, which makes it difficult to predict its stance orientation.

Along with this direction, Fig. 5 shows the confusion matrix for multi-target stance detection.

Table 9 Unsuccessful example tweets with their predicted and true labels.

Example Tweets	Predicted Label	True Label
Target:Atheism Tweet 1: *RT @br_holden: Just say no to superstitious thought in general and religions in particular. #freethinker*	Favor	Against
Target: Climate Change is a Real Concern Tweet 2: *The stock market froze In the summer ?!!*	Favor	None
Target: Hillary Clinton Tweet 3: *@HillarityPress @Amedicinewoman 1.Food: #Benghazi 4 NeverAteAgain 2.Weather: #Benghazi WasFireHot 3.Pets: #Benghazi 4PetsMissThem*	None	Against
Target: Feminist Movement Tweet 4: *Man: this is an issue and needs to be addressed Woman: this is an iss... Men: why are you so outraged???? #everydaysexism*	Against	Favor
Target: Legalization of Abortion Tweet 5: *What about in cases of rape*	Against	Favor

5. Conclusion and Future Directions

In this paper, we have proposed a neural network model, MKC-LSTMVs-ATT, for the target-specific tweet stance detection. Upon extracting higher level feature sequences through multi-kernel convolution, we employed the densely connected Bi-LSTM and nested LSTMs for learning long-term dependencies. Later, a feed-forward attention mechanism is applied to each module to aggregate the final states sequences according to their relative importance. The generated feature sequences are then concatenated and passed to a fully connected layer to estimate the final stance label. Experimental results have shown that our proposed model learn the contextual information effectively which in turns improves the stance detection performance and outperform the state-of-the-art deep learning based methods for both the single and multi-target stance detection benchmark datasets.

In the future, we have a plan to leverage external knowledge to capture the contextual information of the ambiguous, conversational, and extremely short tweets. Moreover, we have a plan to generalize our model for target-independent stance detection in the same domain.

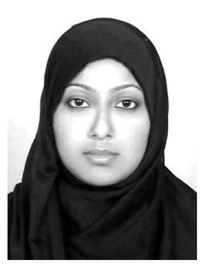
Acknowledgments

The part of this research is supported by MEXT KAKENHI, Grant-in-Aid for Scientific Research (B), Grant Number 17H01746 and the KDDI Foundation.

References

[1] S. Mohammad, S. Kiritchenko, P. Sobhani, X. Zhu, and C. Cherry, "Semeval-2016 task 6: Detecting stance in tweets," Proc. 10th International Workshop on Semantic Evaluation (SemEval), pp.31–41, 2016.
[2] K. Dey, R. Shrivastava, and S. Kaushik, "Topical stance detection for twitter: A two-phase lstm model using attention," Proc. 40th European Conference on Information Retrieval (ECIR), pp.529–536, Springer, 2018.
[3] W. Wei, X. Zhang, X. Liu, W. Chen, and T. Wang, "Pkudblab at semeval-2016 task 6: A specific convolutional neural network system for effective stance detection," Proc. 10th International Workshop on Semantic Evaluation (SemEval), NAACL-HLT, pp.384–388, 2016.
[4] G. Zarrella and A. Marsh, "Mitre at semeval-2016 task 6: Transfer learning for stance detection," arXiv preprint arXiv:1606.03784, 2016.
[5] P. Sobhani, D. Inkpen, and X. Zhu, "A dataset for multi-target stance detection," Proc. 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL): vol.2, Short Papers, pp.551–557, ACL, 2017.
[6] Z. Ding, R. Xia, J. Yu, X. Li, and J. Yang, "Densely connected bidirectional lstm with applications to sentence classification," CCF International Conference on Natural Language Processing and Chinese Computing (NLPCC), pp.278–287, Springer, 2018.
[7] J.R.A. Moniz and D. Krueger, "Nested lstms," Proc. 9th Asian Conference on Machine Learning (ACML), pp.530–544, Springer, 2017.
[8] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from congressional floor-debate transcripts," Proc. 11th Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.327–335, Association for Computational Linguistics, 2006.
[9] S. Somasundaran and J. Wiebe, "Recognizing stances in ideological on-line debates," Proc. NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text (CAAGET), pp.116–124, ACL, 2010.
[10] P. Anand, M. Walker, R. Abbott, J.E.F. Tree, R. Bowmani, and M. Minor, "Cats rule and dogs drool!: Classifying stance in online debate," Proc. 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA), pp.1–9, Association for Computational Linguistics, 2011.
[11] M. Walker, J.F. Tree, P. Anand, R. Abbott, and J. King, "A corpus for research on deliberation and debate," Proc. 8th International Conference on Language Resources and Evaluation (LREC-2012), European Language Resources Association (ELRA), 2012.
[12] K.S. Hasan and V. Ng, "Stance classification of ideological debates: Data, models, features, and constraints," Proc. 6th International Joint Conference on Natural Language Processing (IJCNLP), pp.1348–1356, 2013.
[13] D. Sridhar, L. Getoor, and M. Walker, "Collective stance classification of posts in online debate forums," Proc. Joint Workshop on Social Dynamics and Personal Attributes in Social Media, pp.109–117, 2014.
[14] W. Ferreira and A. Vlachos, "Emergent: A novel data-set for stance classification," Proc. 15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pp.1163–1168, 2016.
[15] P. Sobhani, D. Inkpen, and S. Matwin, "From argumentation mining to stance classification," Proc. 2nd Workshop on Argumentation Mining, Denver, CO, pp.67–77, Association for Computational Linguistics, June 2015.
[16] A. Faulkner, "Automated classification of stance in student essays: An approach using stance target information and the wikipedia link-based measure," Proc. 27th International Florida Artificial Intelligence Research Society (FLAIRS) Conference, pp.174–179, 2014.
[17] A. Rajadesingan and H. Liu, "Identifying users with opposing opinions in twitter debates," Proc. 7th International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction (SBP), pp.153–160, Springer, 2014.
[18] S. Djemili, J. Longhi, C. Marinica, D. Kotzinos, and G.E. Sarfati, "What does twitter have to say about ideology?," NLP 4 CMC: Natural Language Processing for Computer-Mediated Communication/Social Media-Pre-conference workshop at Konvens 2014, Volume-1, Universitätsverlag Hildesheim, 2014.
[19] J. Du, R. Xu, Y. He, and L. Gui, "Stance classification with target-specific neural attention networks," Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI), pp.3988–3994, 2017.
[20] Y. Zhou, A.I. Cristea, and L. Shi, "Connecting targets to tweets: Semantic attention-based model for target-specific stance detection," Proc. 18th International Conference on Web Information Systems Engineering (WISE), pp.18–32, Springer, 2017.
[21] P. Wei, W. Mao, and D. Zeng, "A target-guided neural memory model for stance detection in twitter," Proc. 2018 International Joint Conference on Neural Networks (IJCNN), pp.1–8, IEEE, 2018.
[22] U.A. Siddiqua, A.N. Chy, and M. Aono, "Stance detection on microblog focusing on syntactic tree representation," Proc. 3rd International Conference on Data Mining and Big Data (DMBD), pp.478–490, Springer, 2018.
[23] P. Wei, J. Lin, and W. Mao, "Multi-target stance detection via a dynamic memory-augmented network," Proc. 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp.1229–1232, ACM, 2018.
[24] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Advances in Neural Information Processing Systems (NIPS), pp.3111–3119, 2013.
[25] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," Proc. 19th Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1532–1543, 2014.
[26] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," Transactions of the Association for Computational Linguistics (TACL), vol.5, pp.135–146, 2017.
[27] Y. Kim, "Convolutional neural networks for sentence classification," Proc. 19th Conference on Empirical Methods in Natural Language Processing (EMNLP), pp.1746–1751, ACL, 2014.
[28] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," arXiv preprint arXiv:1510.03820, 2015.
[29] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification," Proc. 26th International Joint Conference on Artificial Intelligence (IJCAI), pp.2915–2921, AAAI Press, 2017.
[30] S. Kim, J.H. Hong, I. Kang, and N. Kwak, "Semantic sentence matching with densely-connected recurrent and co-attentive information," arXiv preprint arXiv:1805.11360, 2018.
[31] D. Li, Y. Chen, M. Gao, S. Jiang, and C. Huang, "Multimodal ges-

ture recognition using densely connected convolution and blstm," Proc. 24th International Conference on Pattern Recognition (ICPR), pp.3365–3370, IEEE, 2018.
[32] C. Wu, F. Wu, Y. Huang, S. Wu, and Z. Yuan, "Thu_ngn at ijcnlp-2017 task 2: Dimensional sentiment analysis for chinese phrases with deep lstm," Proc. 8th International Joint Conference on Natural Language Processing (IJCNLP), Shared Tasks, pp.47–52, 2017.
[33] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol.9, no.8, pp.1735–1780, 1997.
[34] A. Graves, "Generating sequences with recurrent neural networks," arXiv preprint arXiv:1308.0850, 2013.
[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Advances in Neural Information Processing Systems (NIPS), pp.5998–6008, 2017.
[36] S. Fotso, P. Spanoudes, B.C. Ponedel, B. Reynoso, and J. Ko, "Attention fusion networks: Combining behavior and e-mail content to improve customer support," arXiv preprint arXiv:1811.03169, 2018.
[37] Y. Zhang, J. Wang, and X. Zhang, "Ynu-hpcc at semeval-2018 task 1: Bilstm with attention based sentiment analysis for affect in tweets," Proc. 12th International Workshop on Semantic Evaluation (SemEval), pp.273–278, 2018.
[38] C. Raffel and D.P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," arXiv preprint arXiv:1512.08756, 2015.
[39] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[40] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning," University of Toronto, Technical Report, 2012.
[41] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.
[42] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., "Tensorflow: A system for large-scale machine learning," Proc. 12th USENIX Conference on Operating Systems Design and Implementation (OSDI), pp.265–283, USENIX Association, 2016.
[43] J.D. Owens, M. Houston, D. Luebke, S. Green, J.E. Stone, and J.C. Phillips, "Gpu computing," Proc. IEEE, vol.96, no.5, pp.879–899, 2008.
[44] C. Zhou, C. Sun, Z. Liu, and F.C.M. Lau, "A c-lstm neural network for text classification," CoRR, vol.abs/1511.08630, 2015.
[45] E. Loper and S. Bird, "Nltk: The natural language toolkit," Proc. ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics, 2002.
[46] C. Baziotis, N. Pelekis, and C. Doulkeridis, "Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis," Proc. 11th International Workshop on Semantic Evaluation (SemEval), Vancouver, Canada, pp.747–754, ACL, Aug. 2017.
[47] T. Segaran and J. Hammerbacher, Beautiful data: The stories behind elegant data solutions, O'Reilly Media, Inc., 2009.
[48] M. Tutek, I. Sekulic, P. Gombar, I. Paljak, F. Culinovic, F. Boltuzic, M. Karan, D. Alagić, and J. Šnajder, "Takelab at semeval-2016 task 6: Stance classification in tweets using a genetic algorithm based ensemble," Proc. 10th International Workshop on Semantic Evaluation (SemEval), pp.464–468, 2016.
[49] S.M. Mohammad, P. Sobhani, and S. Kiritchenko, "Stance and sentiment in tweets," ACM Transactions on Internet Technology (TOIT), vol.17, no.3, p.26, 2017.
[50] P. Sobhani, D. Inkpen, and X. Zhu, "Exploring deep neural networks for multitarget stance detection," Computational Intelligence (CI), vol.35, no.1, pp.82–97, 2019.



Umme Aymun Siddiqua received her B.Sc. (Engg.) degree from the International Islamic University Chittagong (IIUC), Chittagong, Bangladesh in 2016. She is currently a Master student at the Toyohashi University of Technology, Toyohashi, Japan. Her research interests include opinion mining and knowledge discovery from social media data especially stance detection, sentiment analysis, emotion analysis, and hate speech detection.

Abu Nowshed Chy received his B.Sc. (Hons.) degree from the University of Chittagong, Chittagong, Bangladesh in 2012 and his M.Eng. and Ph.D. degrees from the Toyohashi University of Technology, Toyohashi, Japan in 2016 and 2019. His research interests include microblog search, temporal information retrieval, multimodal information retrieval, opinion mining, natural language processing, and deep learning.

Masaki Aono received the BS and MS degrees from the Department of Information Science from the University of Tokyo, Tokyo, Japan, and the PhD degree from the Department of Computer Science at Rensselaer Polytechnic Institute, New York. He was with the IBM Tokyo Research Laboratory from 1984 to 2003. He is currently a professor at the Graduate School of Computer Science and Engineering Department, Toyohashi University of Technology. His research interests include text and data mining for massive streaming data, and information retrieval for multimedia including 2D images, videos, and 3D shape models. He is a member of the ACM and IEEE Computer Society. He has been a Japanese delegate of the ISO/IEC JTC1 SC24 Standard Committee since 1996.