

Stance Detection with Hierarchical Attention Network

Qingying Sun^{1,3}, Zhongqing Wang¹, Qiaoming Zhu^{1,2} and Guodong Zhou¹

¹School of Computer Science and Technology, Soochow University, Suzhou, China

²Institute of Artificial Intelligence, Soochow University, Suzhou, China

³Huaiyin Normal University, Huaian, China

qingying.sun@foxmail.com, wangzq.antony@gmail.com,
{qmzhu, gdzhou}@suda.edu.cn

Abstract

Stance detection aims to assign a stance label (*for* or *against*) to a post toward a specific target. Recently, there is a growing interest in using neural models to detect stance of documents. Most of these works model the sequence of words to learn document representation. However, much linguistic information, such as polarity and arguments of the document, is correlated with the stance of the document, and can inspire us to explore the stance. Hence, we present a neural model to fully employ various linguistic information to construct the document representation. In addition, since the influences of different linguistic information are different, we propose a hierarchical attention network to weigh the importance of various linguistic information, and learn the mutual attention between the document and the linguistic information. The experimental results on two datasets demonstrate the effectiveness of the proposed hierarchical attention neural model.

1 Introduction

While a lot of works on document-level opinion mining have involved determining the polarity expressed in a customer review (Pang et al., 2008; Liu, 2012), researchers have begun exploring new opinion mining tasks in recent years. One such task is stance detection: given a post written for a two-sided topic discussed in an online debate forum, determine which of the two sides (i.e., *for* and *against*) its author is taking. Stance detection system can potentially have a positive social impact and are of practical interest to non-profits, governmental organizations, and companies.

Previously, some of the related researches used feature engineering to extract either linguistic (Somandaran and Wiebe, 2010) or structure (Sridhar et al., 2015) features manually. Recently, neural models have achieved high success and obtained the best results on stance detection (Zarrella and Marsh, 2016; Du et al., 2017).

A key issue of the neural model is document representation, and most of previous works learn document representation from word sequence using Convolutional Neural Networks (CNNs) (Chen and Ku, 2016; Vijayaraghavan et al., 2016) or Recurrent Neural Networks (RNNs) (Augenstein et al., 2016; Du et al., 2017).

E1: I understand that homosexuals can't have kids. They can adopt children and thus, support the advancement of the human race. (Target: Gay Rights)

Diagram illustrating the sentence: **E1:** I understand that homosexuals can't have kids. They can adopt children and thus, support the advancement of the human race. (Target: Gay Rights)

Dependency arcs are shown:

- nsbj: I → understand
- doobj: understand → that
- nsbj: They → can adopt
- doobj: can adopt → children
- and: children → and
- doobj: thus, support → the advancement of the human race
- nmod:of: the advancement of the human race → race

However, besides the word sequence, stance is correlated with some explicit and implicit linguistic factors. Take E1 for example, both sentimental words (i.e., *understand*, *support*) and argument sentence (i.e., *They can adopt ...*) support the *favor* stance toward the target “Gay Rights”. In addition, the dependency pair, (*adopt*, *children*) and (*advancement*, *race*), also express a positive stance. Hence,

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>

we should learn the document representation by fully employing these linguistic factors (i.e., sentiment, argument, dependency) into neural model.

Moreover, the influence of different linguistic factors of each document is different. For example, the argument sentence in E1 is more important than sentiment and dependency information, and indicates the stance toward the target directly. Hence, we should measure the importance of different linguistic factors for each document through the neural model.

To address the above challenges, we propose a hierarchical attention model, which stands for the mutual attention between the document and the linguistic factors. The model contains two parts: linguistic attention part and hyper attention part. The former helps learn flexible and adequate document representation with different linguistic feature set, and the latter helps adjust the weight of different feature sets. In summary, the contributions are as follows.

- We present a novel hierarchical attention based neural model tailored to stance detection task, which considers the mutual influence between the representation of documents and the corresponding feature sets.
- We make a systematic comparison of LSTM with different linguistic features using the same benchmarks, and we explore influence of different linguistic features on neural models for stance detection.
- Experimental results on two open datasets demonstrate the effectiveness of the proposed approach.

2 Related Works

Previous works on stance detection have focused on congressional debates (Thomas et al., 2006; Burfoot et al., 2011), company-internal discussions (Agrawal et al., 2003), and debates in online forums (Somasundaran and Wiebe, 2010; Anand et al., 2011). Recently, there is a growing interest in performing stance detection on microblogs (Mohammad et al., 2016; Du et al., 2017). Most of existing works on stance detection can be separated into three categories: using linguistic features, using structure features, and using neural models. In the following subsections, we discuss the works on these three categories one by one.

2.1 Linguistic Features

Mosts of previous works employed various kinds of linguistic features for stance detection. For example, Somasundaran and Wiebe (2010) developed a baseline for stance detection by modeling verbs and sentiments. Anand et al. (2011) augmented the n-gram features with lexicon-based and dependency-based features. Except the above transitional linguistic factors, Hasan and Ng (2014) considered the importance of argument for stance detection, and explored the relations between stance and argument, and proposed a pipeline system to predict the stance and extract argument sentence from documents jointly.

2.2 Structure Features

Although linguistic features show effectiveness for stance detection in many works, stance detection has been newly posed as collective classification task with extra structure information. For example, citation structure (Burfoot et al., 2011) or rebuttal links (Walker et al., 2012), was used as extra information to model agreements or disagreements in debate posts and to infer their labels. Moreover, since similar users should express a similar opinion toward the same topic, user information is always used in stance detection. For example, Murakami and Raymond (2010) proposed a maximum cut method to aggregate stances in multiple posts to infer a user’s stance on the target. Rajadesingan and Liu (2014) used a retweet-based label propagation method, which started from a set of opinionated users and labeled tweets by the people who are in the retweet network. Sridhar et al. (2015) utilized Probabilistic Soft Logic (PSL) to collectively classify the stance of users and stance in posts.

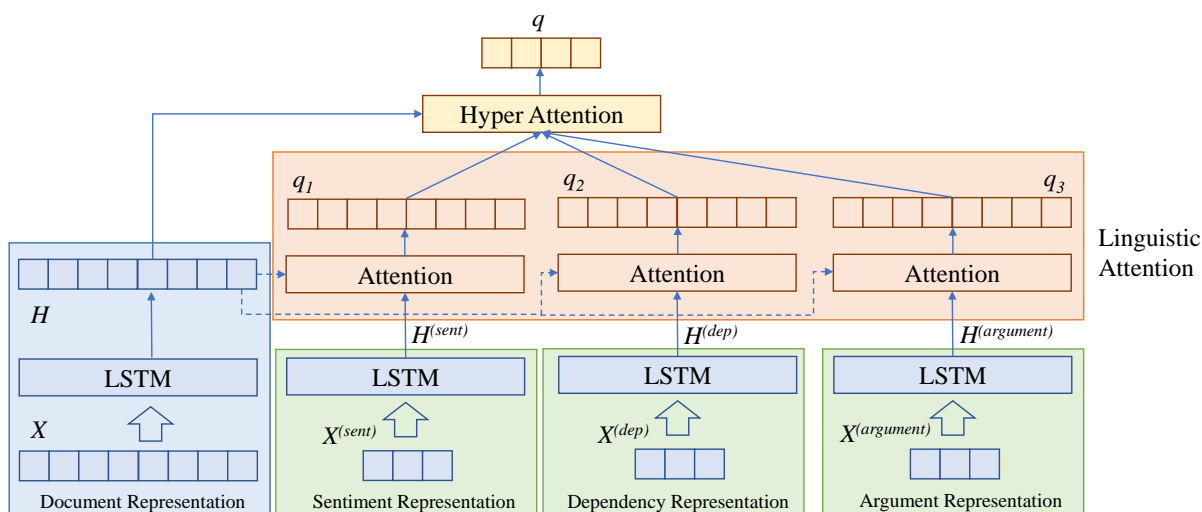


Figure 1: Overview of proposed model.

2.3 Neural Network Models

Since the deep neural network approaches can automatically extract features, there are many works using neural models to detect the stance recently, instead of using various kinds of linguistic features. For example, Augenstein et al. (2016) used two bidirectional RNN to model both target and text for stance detection, and Vijayaraghavan et al. (2016) proposed a combination of classifiers using word-level or character-level CNN models for stance detection. In addition, extra-information, especially user information, is incorporated into neural models for stance detection. For example, Chen and Ku (2016) proposed a user-based neural model to classify stance by using user tastes, topic tastes, and user comments on posts. Du et al. (2017) proposed a neural network-based model, which incorporated target-specific information into stance detection by following a novel attention mechanism.

Different from previous works, which either consider linguistic features individually, or learn the document representation on the word sequence, we propose a neural model to learn mutual attention between the document and the linguistic factors, and achieve best results.

3 Hierarchical Attention Neural Model

Formally, given a natural language document X , the stance detection system returns a binary label Y to denote the stance of the document, where $Y = 1$ denotes *favor* and $Y = 0$ denote *against* toward a special target. The architecture of our proposed hierarchical attention based stance detection system is shown in Figure 1, which illustrates the basic flow of our approach. First, we learn the representation of each document and linguistic features. Then, a hierarchical attention neural network is employed to represent the document under the influence of the different linguistic features: linguistic attention is used to learn the correlations between document representation and different linguistic feature set, and the hyper attention is employed to adjust the weight of different feature sets.

In the following subsections, we firstly show the representation of a document, and then show the representation of different linguistic features. Finally, we illustrate hierarchical attention model, and training process.

3.1 Document Representation

We use a standard Long Short-Term Memory (LSTM) model (Hochreiter and Schmidhuber, 1997) to learn the document representation. LSTM has been proven to be effective in many natural language processing (NLP) tasks such as machine translation (Sutskever et al., 2014) and dependency parsing (Dyer et al., 2015), and it is adept in harnessing long sentences. Let $X = (w_1, w_2, \dots, w_n)$ be a document, where n is the document length and w_i is the i -th token. We transform each token w_i into a real-valued vector x_i using the word embedding vector of w_i , obtained by looking up a pre-trained word embedding

table D . We use the skip-gram algorithm to train embeddings (Mikolov et al., 2013).

The LSTM is used over X to generate a hidden vector sequence (h_1, h_2, \dots, h_n) . At each step t , the hidden vector h_t of LSTM model is computed based on the current vector x_t and the previous vector h_{t-1} , and $h_t = LSTM(x_t, h_{t-1})$. The initial state and all stand LSTM parameters are randomly initialized and tuned during training. We use $H = h_n$ as the representation for X .

3.2 Representation of Linguistic Information

After obtaining the document representation, we learn the representation of different linguistic information.

Sentiment Representation

In principle, sentiment information of a post highly influences the stance. For example, the author is in favor of the target in E2, since he expresses the positive opinion toward the target. In addition, the stance toward the target and the sentiment of the post may be opposite in some posts. As in E3, the author is in favor of the given target although the post expresses the negative opinion.

E2: Hillary is our best choice if we truly want to continue being a progressive nation. (*Target: Hillary Clinton*)

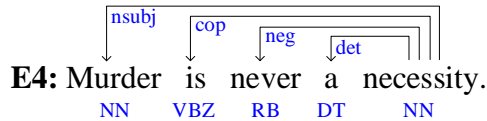
E3: When the last tree is cut down, the last fish eaten & the last stream poisoned, you will realize that you cannot eat money. (*Target: Climate Change is a Real Concern*)

Hence, it is a challenge task to employ sentiment information for stance detection, since the polarity of a post could be either the same or opposite toward the stance. In this study, we do not integrate sentiment information into document representation directly, but use attention mechanism to measure the correlation between document representation and sentiment information.

Particularly, we simply extract the sentimental word sequence $X^{(sent)} = \{x_1^s, x_2^s, \dots, x_m^s\}$ of each document from sentiment lexicon, where x_i^s means sentiment word. We then learn the representation of sentiment information through this sentimental word sequence using LSTM model, and use $H^{(sent)} = LSTM(x_m^s, h_{m-1})$ as sentiment representation.

Dependency Representation

The dependency-based features can be utilized to capture the inter-word relationships (Anand et al., 2011; Persing and Ng, 2016). The dependency structure involves some stance-taking text related to the given target. As in E4, we notice that the important words that express a stance in the sentence are “murder”, “never” and “necessity”. By analyzing the dependency structure in this and other prompt parts, we can extract the stance-taking dependency information we identify as “murder-never-necessity”.



Hence, we extract the pair of arguments sequence $X^{(dep)} = \{x_1^d, x_2^d, \dots, x_m^d\} = \{x_1 \oplus x_3, \dots, x_i \oplus x_j\}$ involved in each dependency relation from a dependency parser as a feature, where $x_i^d = x_j \oplus x_k$ is arguments pair from dependency relation.

We then learn the representation of dependency information through the dependency sequence $X^{(dep)}$ using LSTM model, and we use $H^{(dep)} = LSTM(x_m^d, h_{m-1})$ as dependency representation.

Argument Representation

In general, there are not only the stance of the author toward the target in a post, but also personal beliefs about what is true or what action should be taken. This personal belief is called argument (Wilson and Wiebe, 2005). There exist some arguments behind the stance people express on a specified target. If we can extract the argument sentence from the post, then it will be beneficial to detect the author’s stance.

As in E5, with the help of argument sentence with bold, it will be easy to detect that the author support the legalization of marijuana.

E5: Most issues like this, such as sex between minors and alcohol, come down to one thing: it's your choice. If you want to ruin your life, be my guest. It isn't the government's job to control that. (*Target: Legalization of marijuana*)

In order to utilize argument information for stance detection, we extract the argument sentence from each document, and treat argument sentence detection as a binary classification task (Hasan and Ng, 2014). After we get the argument sequence $X^{(argument)} = \{x_1^a, x_2^a, \dots, x_m^a\}$ from argument sentences, where x_i^a is the i -th word in the argument sentences. We then learn the representation of argument information through the argument sequence $X^{(argument)}$ using LSTM model, and we use $H^{(argument)} = LSTM(x_m^a, h_{m-1})$ as argument representation.

3.3 Linguistic Attention

Based on our assumption, each linguistic resource should focus on different words of the same document. The extent of attention can be measured by the relatedness between each word representation $h_j \in H$ and a linguistic representation $l_i \in \{H^{(sent)}, H^{(dep)}, H^{(argument)}\}$. We propose the following formulas to calculate the weights.

$$\alpha_{ij} = \frac{\exp(w_{ij})}{\sum_{j=1}^n \exp(w_{ij})} \quad (1)$$

$$w_{ij} = \tanh(W^T [h_j : l_i] + b) \quad (2)$$

Here, α_{ij} denotes the weight of attention from a feature set l_i to the j th word in the document. \tanh is a non-linear activation function. W is an intermediate matrix, and b is the offset. Both of them are randomly initialized and updated during training. Subsequently, according to the specific linguistic feature l_i , the attention weights are employed to calculate a weighted sum of the hidden representations, resulting in a semantic vector q_i that represents the document with the corresponding feature set.

$$q_i = \sum_{j=1}^n \alpha_{ij} h_j \quad (3)$$

3.4 Hyper Attention

Intuitively, different documents should value the three linguistic feature set differently. We thus define the final representation by weighting document representation and different linguistic representations $\{H, q_1, q_2, q_3\}$, as follows.

$$q = \sum_{j=1}^4 \beta_j k_j, \quad k_j \in \{H, q_1, q_2, q_3\} \quad (4)$$

$$\beta_j = \frac{\exp(w_j)}{\sum_{j=1}^4 \exp(w_j)} \quad (5)$$

$$w_j = \tanh(W^T V_j + b) \quad (6)$$

Here β_j denotes the attention of different document representation, indicating which document representation should be more focused. W is also an intermediate matrix, V is the weight matrix of different document representation and b is an offset value. Both of them are randomly initialized and updated during training. We use q as the final representation to learn and predict the model.

| Stance | Abortion | GayRights | Obama | Marijuana |
|-------------|----------|-----------|-------|-----------|
| Favor (%) | 54.9 | 63.4 | 53.9 | 69.5 |
| Against (%) | 45.1 | 36.6 | 46.1 | 30.5 |
| Total | 1741 | 1376 | 985 | 626 |

Table 1: Distribution of H&N14 dataset.

| Stance | Atheism | Climate | Feminism | Hillary | Abortion |
|-------------|---------|---------|----------|---------|----------|
| Favor (%) | 16.9 | 59.4 | 28.2 | 16.6 | 17.9 |
| Against (%) | 63.3 | 4.6 | 53.9 | 57.4 | 58.3 |
| None (%) | 19.8 | 36.0 | 17.9 | 26.0 | 23.8 |
| Total | 733 | 564 | 949 | 984 | 933 |

Table 2: Distribution of SemEval16 dataset.

3.5 Training

We employ cross-entropy loss function to train the model. Specially, the loss function is defined as follows:

$$L(\Theta) = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log p_{ij} + \frac{\lambda}{2} \|\Theta\|^2 \quad (7)$$

where n is the total number of items of training data; m is the number of category; y_{ij} indicates whether the i -th sample belongs to the j -th category, which is the ground-truth label; and p_{ij} represents the predicted probability. Θ is the set of model parameters and λ is a parameter for L2 regularization.

Standard back propagation is performed to optimize parameters. We employ Adam (Kingma and Ba, 2014) for parameter optimization and Word embeddings are trained using the *Skip-gram* algorithm (Mikolov et al., 2013)¹.

4 Experiments

4.1 Datasets and Preprocessing

We use two datasets to evaluate the performance of the proposed system. **H&N14** is collected by Hasan and Ng (2014), and **SemEval16** is from SemEval-2016 Share Task 6.A (Mohammad et al., 2016).

H&N14 is collected from an English online debate forum with four targets: “Abortion”, “Gay Rights”, “Obama”, and “Marijuana”. The distribution of the dataset is shown in Table 1. In Hasan and Ng (2014)’s setting, the dataset is split into five folds, and they conduct five-fold cross-validation on the division. For fair comparison, we follow their setting in all experiments.

SemEval16 is the dataset for stance detection from English tweets, and each tweet corresponds to a special target: “Atheism”, “Climate Change is a Real Concern”(“Climate”), “Feminist Movement”(“Feminist”), “Hillary Clinton”(“Hillary”), and “Legalization of Abortion”(“Abortion”). The distribution of the dataset is shown in Table 2. The dataset has already been separated into training and testing set (Mohammad et al., 2016).

The average length of documents in H&N14 and SemEval16 is 114 and 18, respectively. The length of document in SemEval16 is much shorter than H&N14, since SemEval16 is collected from tweets. Based on such difference of source and document length, we can evaluate the performance of the proposed model on a different setting.

Some preprocessing approaches are employed to extract sentiment, dependency, and argument information from the two datasets:

- The sentiment word sequence of each document is extracted from MPQA subjective lexicon².

¹<https://code.google.com/p/word2vec/>

²http://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

- The dependency sequence is extracted from each sentence by using Stanford Parser³. We select the relations which include noun, verb, adjective, adverb or negation words such as “nsubj”, “acl”, “dobj” etc. and get the argument pairs in these relations.
- The argument sentence is extracted from each document. We firstly separate the document into a sentence set based on punctuation and connective, and then we use a SVM classifier to predict the argument label of each sentence. We employ the annotated training set in (Hasan and Ng, 2014) and simplify the argument sentence detection as a binary classification task which means that sentence is just identified whether or not an argument of the related stance specifically.

Hyper-parameters

We train 300 dimensional word embedding using Word2Vec (Mikolov et al., 2013) on all the training data, and fine-tuning during the training process. The dimension of LSTM layers and the output of attention mechanism is set to 300. The dropout rate is set to 0.25. The other hyper-parameters and learning rate are tuned on the development data. The development data are extracted from 10% of training data.

4.2 Evaluation Metrics

Both micro and macro average of F1-score across topics are adopted as the metrics (Mohammad et al., 2016). F1-score for Favor and Against categories of all instances is calculated as:

$$F_{favor} = \frac{2 \times P_{favor} \times R_{favor}}{P_{favor} + R_{favor}} \quad (8)$$

$$F_{against} = \frac{2 \times P_{against} \times R_{against}}{P_{against} + R_{against}} \quad (9)$$

where P and R is precision and recall. Then the average of F_{Favor} and $F_{Against}$ is calculated as the final metrics:

$$F_{avg} = \frac{F_{favor} + F_{against}}{2} \quad (10)$$

We average the F_{avg} on each topic to calculate macro-average F score ($MacF_{avg}$). Besides, we calculate F_{avg} score across all targets to get micro-average F score ($MicF_{avg}$).

Note that, different from H&N14, which only contains *favor* and *against* posts, SemEval16 dataset contains *none* posts, which do not express any stance. However, the final metrics disregard the None class. Following Mohammad et al. (2016) and Du et al. (2017), we take the average F-score for Favor and Against classes, and treat None as a class that is not of interest.

4.3 Experiment Results

Comparison with Baselines

We compare our proposed model with the following state-of-the-art baseline systems:

- *SVM* is a baseline model in many previous works (Hasan and Ng, 2014; Mohammad et al., 2016). We use libSVM⁴ to train the SVM model with bag-of-words features.
- *LSTM* employs word embedding, and learn the document representation through LSTM model.
- *TAN* is proposed by Du et al. (2017), it uses an attention mechanism to learn the correlation between topic and document representation. It reports the best results on SemEval16 dataset.
- *HAN* is our proposed model, which uses Hierarchical Attention Neural (HAN) model to learn the mutual attention between document and linguistic information.

| Model | Abortion | GayRights | Obama | Marijuana | $MacF_{avg}$ | $MicF_{avg}$ |
|-------|--------------|--------------|--------------|--------------|--------------|--------------|
| SVM | 59.48 | 59.52 | 63.02 | 55.02 | 59.26 | 60.52 |
| LSTM | 60.72 | 56.07 | 60.14 | 55.58 | 58.13 | 59.45 |
| TAN | 63.96 | 58.13 | 63.00 | 56.88 | 60.49 | 62.35 |
| HAN | 63.66 | 57.36 | 65.67 | 62.03 | 62.18 | 63.25 |

Table 3: Comparison with baselines on H&N14 dataset.

| Model | Atheism | Climate | Feminism | Hillary | Abortion | $MacF_{avg}$ | $MicF_{avg}$ |
|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| SVM | 62.16 | 42.91 | 56.43 | 55.68 | 60.38 | 55.51 | 67.01 |
| LSTM | 58.18 | 40.05 | 49.06 | 61.84 | 51.03 | 52.03 | 63.21 |
| TAN | 59.33 | 53.59 | 55.77 | 65.38 | 63.72 | 59.56 | 68.79 |
| HAN | 70.53 | 49.56 | 57.50 | 61.23 | 66.16 | 61.00 | 69.79 |

Table 4: Comparison with baselines on SemEval16 dataset.

| Model | Abortion | GayRights | Obama | Marijuana | $MacF_{avg}$ | $MicF_{avg}$ |
|---------------------|----------|-----------|-------|-----------|--------------|--------------|
| SVM | 59.48 | 59.52 | 63.02 | 55.02 | 59.26 | 60.52 |
| SVM+Ling | 62.17 | 58.69 | 65.14 | 56.56 | 60.64 | 62.25 |
| LSTM | 60.72 | 56.07 | 60.14 | 55.58 | 58.13 | 59.45 |
| LSTM+Att+Sentiment | 63.88 | 58.69 | 63.76 | 57.08 | 60.85 | 62.45 |
| LSTM+Att+Dependency | 63.15 | 59.48 | 64.69 | 59.21 | 61.63 | 62.67 |
| LSTM+Att+Argument | 62.82 | 58.31 | 64.50 | 61.91 | 61.89 | 63.14 |
| HAN | 63.66 | 57.36 | 65.67 | 62.03 | 62.18 | 63.25 |

Table 5: Influence of linguistic feature.

We compare the proposed HAN model with baseline systems on H&N14 and SemEval16 datasets, and the results are given in Table 3 and Table 4. From the results, we can find that SVM outperforms LSTM in these two datasets. It indicates that, simply document representation on word sequence cannot capture useful information for stance detection. TAN outperforms both SVM and LSTM, and it indicates that target information is helpful for stance detection. Moreover, our proposed model outperforms all the discrete and neural models. It shows the effectiveness of linguistic features and proposed hierarchical attention model.

4.4 Influence of Linguistic Features

We then analyze the influence of proposed different linguistic features on H&N14 dataset. The results are shown in Table 5, where *SVM+Ling* is SVM classification with all the linguistic features employed in our paper, and *LSTM+Att+** employs attention mechanism to consider the correlations between document representation and each linguistic representation individually. For example, *LSTM+Att+Sentiment* means the model just employs linguistic attention of sentiment information in Section 3.3.

From the results, we can find that SVM+Ling outperforms SVM, it proves the effectiveness of linguistic information directly. Moreover, we find that every linguistic information with attention mechanism is effective for stance detection. Especially, the argument information outperforms other features, it may be due to that argument information is highly related with stance than other linguistic information. In addition, the proposed model with all kinds of linguistic information outperforms all other models which consider each linguistic information individually.

³<https://nlp.stanford.edu/software/lex-parser.shtml>

⁴<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

| Model | Abortion | GayRights | Obama | Marijuana | $MacF_{avg}$ | $MicF_{avg}$ |
|--------------|----------|-----------|-------|-----------|--------------|--------------|
| HAN | 63.66 | 57.36 | 65.67 | 62.03 | 62.18 | 63.25 |
| -Hyper | 62.84 | 57.63 | 64.71 | 58.61 | 60.95 | 62.45 |
| -Hyper,-Ling | 62.33 | 56.96 | 63.26 | 57.60 | 60.04 | 61.50 |
| LSTM | 60.72 | 56.07 | 60.14 | 55.58 | 58.13 | 59.45 |

Table 6: Influence of network configuration.

4.5 Influence of Network Configurations

We study the influence of various network configurations by performing ablation experiments on H&N14 dataset, as shown in Table 6. *-Hyper* denotes ablation of the hyper attention layer, and concatenate $\{H, q_1, q_2, q_3\}$ directly and put them into a fully-connected layer. *-Hyper,-Ling* denotes ablation both of the hyper attention and linguistic attention layer, and concatenates $\{H, H^{(sent)}, H^{(dep)}, H^{(argument)}\}$ together into a fully-connected layer.

By comparing between HAN and “-Hyper”, we can find that hyper attention over all the linguistic representation does have significant influence on the results. Comparison between “-Hyper” and “-Hyper,-Ling”, we can find that linguistic attentions with each linguistic features does have significant improvement. On the other hand, using LSTM to detect stance (LSTM) does not obtain a better performance compared to concatenating linguistic information directly (“-Hyper,-Ling”). This confirms our intuition that stance of document is influenced by various linguistic information.

5 Conclusion

In this paper, we focus on stance detection task. We consider the impacts of different linguistic features when representing the document, and propose a novel hierarchical attention model for stance detection. Specifically, we employ a neural model to represent the documents and their linguistic features with attention mechanism. In addition, on the top of different document representation, we use an attention model to measure the importance of different linguistic features, and learn the mutual attention between the document and the linguistic information. The extensive experiments demonstrate that the proposed model achieves better performance compared with the state-of-the-art models on two datasets.

Acknowledgements

The corresponding author is Zhongqing Wang. We are grateful for the help of Jingjing Wang and Dr. Zhenghua Li for their initial discussion. We thank our anonymous reviewers for their constructive comments, which helped to improve the paper. This work is supported by the National Natural Science Foundation of China (61331011, 61751206, 61773276) and Jiangsu Provincial Science and Technology Plan (No. BK20151222).

References

- Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. 2003. Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th International World Wide Web Conference*, pages 529–535.
- Pranav Anand, Marilyn A. Walker, Rob Abbott, Jean E. Fox Tree, Robeson Bowmani, and Michael Minor. 2011. Cats rule and dogs drool!: Classifying stance in online debate. In *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 1–9.
- Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 876–885.
- Clinton Burfoot, Steven Bird, and Timothy Baldwin. 2011. Collective classification of congressional floor-debate transcripts. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, pages 1506–1515.

- Wei-Fan Chen and Lun-Wei Ku. 2016. UTCNN: a deep learning model of stance classification on social media text. In *26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers*, pages 1635–1645.
- Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 3988–3994.
- Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 334–343.
- Kazi Saidul Hasan and Vincent Ng. 2014. Why are you taking this stance? identifying and classifying reasons in ideological debates. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 751–762.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *27th Annual Conference on Neural Information Processing Systems*, pages 3111–3119.
- Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiao-Dan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, pages 31–41.
- Akiko Murakami and Rudy Raymond. 2010. Support or oppose? classifying positions in online debates from reply activities and opinion expressions. In *23rd International Conference on Computational Linguistics*, pages 869–875.
- Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.
- Isaac Persing and Vincent Ng. 2016. Modeling stance in student essays. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2174–2184.
- Ashwin Rajadesingan and Huan Liu. 2014. Identifying users with opposing opinions in twitter debates. In *Social Computing, Behavioral-Cultural Modeling and Prediction - 7th International Conference*, pages 153–160.
- Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124.
- Dhanya Sridhar, James R. Foulds, Bert Huang, Lise Getoor, and Marilyn A. Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, pages 116–125.
- Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.
- Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335.
- Prashanth Vijayaraghavan, Soroush Vosoughi, and Deb Roy. 2016. Automatic detection and categorization of election-related tweets. In *Proceedings of the 10th International Conference on Web and Social Media*, pages 703–706.
- Marilyn A. Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics*, pages 592–596.

Theresa Wilson and Janyce Wiebe. 2005. Annotating attributions and private states. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 53–60.

Guido Zarrella and Amy Marsh. 2016. MITRE at semeval-2016 task 6: Transfer learning for stance detection. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 458–463.