Pro/Con: Neural Detection of Stance in Argumentative Opinion

Marjan Hosseinia, Eduard Dragut and Arjun Mukherjee

April 5, 2019

Pro/Con: Neural Detection of Stance in Argumentative Opinions

Marjan Hosseinia[1], Eduard Dragut[2], and Arjun Mukherjee[1]

[1] University of Houston, Houston TX , USA
mhosseinia@uh.edu, arjun@cs.uh.edu
[2] Temple University, Philadelphia, PA ,USA
edragut@temple.edu

Abstract. Accurate information from both sides of the contemporary issues is known to be an 'antidote in confirmation bias'. While these types of information help the educators to improve their vital skills including critical thinking and open-mindedness, they are relatively rare and hard to find online. With the well-researched argumentative opinions (arguments) on controversial issues shared by Procon.org in a nonpartisan format, detecting the stance of arguments is a crucial step to automate organizing such resources. We use a universal pretrained language model with weight-dropped LSTM neural network to leverage the context of an argument for stance detection on the proposed dataset. Experimental results show that the dataset is challenging, however, utilizing the pretrained language model fine-tuned on context information yields a general model that beats the competitive baselines. We also provide analysis to find the informative segments of an argument to our stance detection model and investigate the relationship between the sentiment of an argument with its stance.

Keywords: stance detection · Universal Language Model Fine-tuning · AWD-LSTM

1 Introduction

The problem of stance detection is to identify whether a given opinion supports an idea or contradicts it. It is relatively new in the area of opinion mining and is recently being explored by more researchers [1–4, 7, 12]. Table 1 provides two arguments. The arguments answer a question while taking a stance of the two possible sides against a controversial issue. A stance that supports an issue is a *pro*, and the other side that is against it is a *con*.

In opinion mining identifying a stance of an opinion is a more challenging task than sentiment analysis [4] and naturally differs from it. Here, the problem is no longer finding the whole polarity of an opinion but is to identify its polarity against an issue. Recently, the argumentative opinions of controversial issues have attracted more people who want to take a stance after seeking enough information about the reason behind opinions from both sides. For example, one

Table 1. Tow arguments, a pro and a con, for the issue ***medical marijuana***. The question is: ***"Should Marijuana Be a Medical Option?"***. Each example is a tuple of type (issue, question, context, argument).

Issue : Medical marijuana
Question : Should Marijuana Be a Medical Option?
Context : In 1970, the US Congress placed marijuana in Schedule I of the Controlled Substances ... Proponents of medical marijuana argue that it can be a safe and effective treatment for the symptoms of cancer, AIDS, multiple sclerosis, pain, glaucoma, epilepsy, and other conditions... Opponents of medical marijuana argue that it is too dangerous to use, lacks FDA-approval, and that various legal drugs make marijuana use unnecessary.

Argument (Pro): Ultimately, the issue is not about laws, science or politics, but sick patients. Making no distinction between individuals circumstances of use, the war on drugs has also become a war on suffering people. Legislators are not health care professionals and patients are not criminals, yet health and law become entwined in a needlessly cruel and sometimes deadly dance... I sincerely hope our work will illuminate the irrational injustice of medical marijuana prohibition.

Argument (Con): We can't really call marijuana medicine. It's not a legitimate medicine. The brain is not fully developed until we're about 25. That's just the way it is, and using any kind of mind-altering substance impacts that development. It needs to go through the FDA process...

might wonder *"Should Marijuana Be a Medical Option?"*. This question might be found in many online debate forums and people who like to consume marijuana or the ones who hate it take a stance without bringing an acceptable justification. These types of opinions are usually short and express the stance directly (e.g. tweets). However, argumentative opinions are generally long, more complex, contain high-level ideas, and take a stance while bringing some reasons. Finding the stance of an argument is not straightforward compared to opinions with spontaneous language (e.g. tweets). See Table 1-pro as an example. We study the problem of stance detection in argumentative opinions of 46 different controversial issues. The arguments are collected and represented in a nonpartisan way which means that they are not biased specifically towards any party.

We make the following contributions in this paper. First, we propose a new stance detection dataset from ProCon[3], a collection of critical controversial issues. Each entity of our ProCon dataset is a tuple of type (issue, question, context, argument) where an *issue* refers to the underlying domain, a *question* asks for an opinion, *context* brings a summary of proponent and opponent viewpoints about the *issue*, and an *argument* is a reason-based opinion for or against the *issue*. Table 2 shows how people justify/condemn "legalization of abortion" while bringing some reasons.

We, also, propose a model that leverages the context of an issue to predict the stance of the given opinion. In ProCon dataset, the average number of opinions per issue per class is 24. This size of data may not be large enough for training a neural network. To compensate for this small size of data we build our model on

[3] https://www.procon.org/

Table 2. Two arguments about "legalization of abortion".

pro	con
The US Supreme Court has declared abortion to be a "fundamental right" guaranteed by the US Constitution. ... decision stated that the Constitution gives "a guarantee of certain areas or zones of privacy," and that "This right of privacy... is broad enough to encompass a woman's decision whether or not to terminate her pregnancy."	Unborn babies are considered human beings by the US government. The federal Unborn Victims of Violence Act, which was enacted "to protect unborn children from assault and murder", states that under federal law, anybody intentionally killing or attempting to kill an unborn child should be punished...for intentionally killing or attempting to kill a human being.

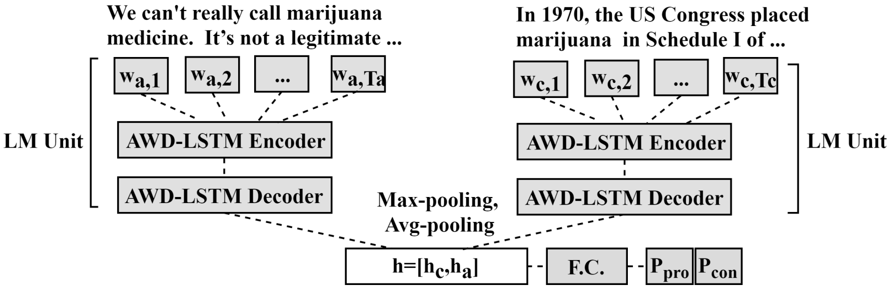

Fig. 1. The model; $w_{a,i}, w_{c,i}$ ith word of argument and context sequence; F.C.: Fully Connected layer; p_{pro}, p_{con} : class probabilities; dark boxes use pretrained weights.

top of the Universal pretrained Language Model, ULMFiT, [8] and fine tune it to our stance detection task. The pretrained language model is a "counterpart of ImageNet for NLP" and can be used in various tasks independent of document size and label as well as the number of in-domain documents [8].

The model is detailed in Section 4. We now discuss the related works. Then, we describe our dataset, the proposed model, and the experimental study.

2 Related Works

Most works on stance classification focus on online debates including posts and tweets while typically restricting the underlying data to a few (up to 8) targets (issues). A probabilistic approach that models stance, the target of stance and sentiment of a tweet is trained on a large dataset of around 3K and evaluated on more than 1.2K tweets toward five targets [5]. Hasan et. al. explore a new task of reason classification by modeling stances and reasons on a corpus of ideological debate posts from five domains [7]. They show that developed models of stances and reasons provide better reason and stance classification results than their simpler models. SemEval2016 provides two different stance detection frameworks for tweets [11, 4]: one supervised framework with five targets containing 4K tweets and another weakly supervised with one target [1]. Unlike others, Bar et. al. propose a contrast detection algorithm for 55 different topics [2]. It is designed to detect the stance of *claims* that are defined as *"general, concise statements*

Table 3. ProCon dataset statistics. Docs refer to argumentative opinions (arguments).

	train		dev		test		train	
	docs	docs/issue	docs	docs/issue	docs	docs/issue	words/arg	words/cntx
size	1517	33	178	4	530	12	166 ± 65	177 ± 34

that directly support or contest the given topic". Actually, claims are *"often only a small part of a single Wikipedia sentence"* in their dataset [9]. While all these works have made important contributions, they do not address the problem of detecting stance in fluid and long arguments, which is the focus of this work.

3 Dataset

We collect the information of 46 controversial issues from ProCon, a top-rated nonprofit organization that provides professionally-researched pros and cons to create our dataset (Table 3) [4]. We define each instance as a tuple of type I=(issue, context, question, argument) where the *issue* is a general topic, the *context* introduces the issue and brings a summary of proponent and opponent opinions and an *argument* is a reason-based opinion taking a stance on or against the given *issue* (Table 1). The issues cover various topics from health and medicine, education, politics, science and technology to entertainment and sports. We will use the words *target* and *issue* interchangeably in this paper as *target* convey same meaning in other research. *Argument* supports a position with powerful and compelling statements. The dataset is divided into $1,517$ train, 178 dev, and 530 test samples (Table 3).

4 Model

Inspired by ULMFiT [8], we propose a model to handle both diverse and small training data per issue (Figure 1). Our model has three units: a) parallel Language Model (LM) units to learn an argument and the context of its underlying issue. b) one fusion unit that summarizes all elements of the data and c) the classification unit that predicts the stance. We describe them below.

4.1 Parallel LM units

We let the model jointly learn an argument with its corresponding context using two LM units. A context usually covers a few sentences introducing the issue and two summaries of proponent and opponent arguments (Table1-context). We hypothesize that pro-arguments and con-arguments are related to disjoint parts of the context because of the intrinsic contradiction of pro- and con-arguments. Let $P = ([w_{1,a}, ..., w_{T_a,a}], [w_{1,c}, ..., w_{T_c,c}])$ be input pair where $w_{i,a}$ and $w_{i,c}$ are

[4] For more details visit https://www.procon.org/faqs.php

the ith word of an argument and its context sequence respectively and T_a, T_c are the last time steps. Each LM unit is a three-layer neural network (Figure 1). First, words are represented as vectors of size $d_e = 400$ using the embedding matrix W_e. The matrix is the result of pretraining the Language Model on Wikitext data with more than $103M$ words [10]. Then, a weight-dropped LSTM (AWD-LSTM) encodes word embedding to a higher dimension $(1, 150)$, and another AWD-LSTM decodes the hidden representation of words into the embedding dimension and predicts the next word of the sequence. AWD-LSTM applies recurrent regularization on the hidden-to-hidden weight matrices to prevent over-fitting across its connections. It adds Activation Regularization (AR) and Temporal Activation Regularization (TAR) to the loss function [10]. Later we provides more details of the two regularization techniques. The argument LM unit is the following:

$$\begin{aligned} x_{i,a} &= W_e w_{i,a}, \\ z_{i,a} &= \text{lstm}_{enc,a}(x_{i,a}), i \in [1, T_a], \\ h_{i,a} &= \text{lstm}_{dec,a}(z_{i,a}), i \in [1, T_a] \end{aligned} \tag{1}$$

where $z_{i,a}, h_{i,a}$ are the hidden state of LSTM encoder and decoder respectively. Similarly, $h_{i,c}$ is the output of context LM unit.

4.2 Fusion and Classification

The fusion layer leverages the information of both LM outputs. Most information of an argument is hidden in the last hidden state of the LSTM decoder of the LM unit. However, important information might be hidden anywhere in a *long* document. We use max-pooling and average-pooling of both inputs (argument, context) along with the last hidden state of LSTM decoder for fusion.

$$\begin{aligned} h_a &= [h_{T_a,a}, \text{max-pool}(h_{T_a,a}), \text{avg-pool}(h_{T_a,a})], \\ h_c &= [h_{T_c,c}, \text{max-pool}(h_{T_c,c}), \text{avg-pool}(h_{T_c,c})] \end{aligned} \tag{2}$$

where $h_{T_a,a}$,$h_{T_c,c}$ are the hidden state of LSTM decoder of argument and context LM units at time T_a, T_c and $[,]$ is concatenation. Finally, the pooled information, $h = [h_a, h_c]$, builds the fusion layer and connects an argument with any significant parts of the context. We feed h through a fully connected layer with $d_r = 50$ hidden neurons activated with a rectifier. The second fully-connected layer but with the linear activation gives us 2d vectors to be used by a softmax function for classification. We apply batch-normalization and dropout to both fully-connected layers to avoid over-fitting. As we mentioned earlier, AWD-LSTM adds TAR (l_{tar}) and AR (l_{ar}) to the final loss. AR is an L2-regularization that controls the norm of the weights to reduce over-fitting. And TAR acts as L2 decay and is used on individual activations. It considers the difference of the outputs of the LSTM decoder at consecutive time steps:

$$\begin{aligned} & l_{ar} = \alpha * ||[h_{T_a,a}, h_{T_c,c}]||_2, l_{tar} = \beta * ||[h^{'}_{T_a,a}, h^{'}_{T_c,c}] - [h^{'}_{T_a-1,a}, h^{'}_{T_c-1,c}]||_2 \\ & L = -\sum_{d} \log h_{s,j} + l_{ar} + l_{tar} \end{aligned} \tag{3}$$

where j is the label of the document and $\alpha = 2$, $\beta = 1$ are the scaling coefficients. $h^{'}_{T_a,a}$, $h^{'}_{T_c,c}$ are the last hidden states of the two LSTM decoders without dropout.

5 Evaluation

We compare our model with state-of-the-art methods in stance detection. The methods are as follows:

- BoW-s : is a Bag of Words model that gains the best performance in TaskA of SemEval2016 [12] with SVM classifier. The features are boolean representation (0/1) of word uni-, bi- and tri-grams as well as character 2, 3, 4 and 5-grams. The presence/absence of any manually selected keywords of the underlying issue is also added to the feature vector. For example, for the issue of 'Hillary Clinton' the presence of *Hillary* or *Clinton* sets this feature to true. We manually select at least three keywords per issue in Procon dataset. Unlike [12], we do not build an individual classifier for each issue separately. We create one general classifier trained on the whole dataset. We examine the BoW-s feature vectors with SVM, Gaussian Naive Bayes (GNB), Logistic Regression (LR), and Random Forest (RF).[5]
- Independent Encoding (IE) : is one of the baselines reported in [1]. It learns the representation of a document and its target independently using two parellel LSTMs. Then, the last hidden states of the two LSTMs are concatenated and projected with the *tanh* function. Finally, a softmax predicts the class distribution over the non-linear projection.
- ULMFiT: is the backbone of our model [8]. We keep all settings intact and train the model by applying the discriminative fine-tuning technique.
- Bidirectional Conditional Encoding (BiCoEn): outperforms the existing methods of SemEval 2016-TaskB. In TaskB the goal is to predict the stance of a tweet over one single unseen target, 'Donald Trump'[1]. The model takes a *tweet* and its underlying *target* and initializes the state of the bidirectional LSTM of tweets with the last hidden state of the forward and backward encoding of the target. In this way the model builds target-dependent representations of a tweet while both the left and right sides of a word are considered. This model takes a document (tweet) and its target (e.g. 'Climate Change is a Real Concern' or 'Atheism') as input for training. To make the comparison more reliable we examine BiConEn for both types of input: (*argument*, *context*) and (*argument*, *issue*). Here, *issue* is the *target* as in [1].[6]

6 Results and Analysis

We apply discriminative fine-tuning for ULMFiT and our model. We execute the evaluation 5 times and report the average results for all methods. Table 4

[5] we use scikit-learn with default settings

[6] We use their code shared on https://github.com/sheffieldnlp/stance-conditional

Table 4. Procon dataset results. arg: argument, cntx: context, P:Precision, R:Recall

Method	Input	Pro			Con			**Macro-F1**	**Acc**
		P	R	F1	P	R	F1		
BoW-s+SVM	arg	61	63	62	62	61	61	62	62
BoW-s+RF	arg	64	67	65	66	62	64	65	65
BoW-s+LR	arg	61	63	62	62	60	61	61	61
BoW-s+GNB	arg	58	65	62	61	54	57	59	59
ULMFiT	(arg, cntx)	65.8	61.2	63.4	64	**68.5**	**66.2**	64.8	64.9
IE	(arg, issue)	55.4	60.5	57.5	56.7	51.1	53.2	55.4	56.7
IE	(arg, cntx)	56.9	52.7	54.5	56.1	60.1	57.8	56.2	57.3
BiCoEn	(arg, issue)	56.5	57.7	57	57.2	55.9	56.4	56.7	56.9
BiCoEn	(arg, cntx)	55.9	57.6	56.4	56.7	54.6	55.2	55.8	56.6
Our model	(arg, cntx)	**65.9**	**82.6**	**73.3**	**77**	57.7	65.9	**69.6**	**70.1**

provides the experimental results. The largest values are highlighted in bold and the second largest are underlined. According to the table, both accuracy and Macro-F1 of all baselines do not exceed 65%, showing that the presence of diverse issues makes the problem hard to solve. It is expected that the Neural Network (NN) baselines give weak results compared to BoW for ProCon data. With $1,517$ training samples and 46 different issues, the average number of arguments per issue is 33 which is not enough for fitting NN models unless we provide some external knowledge for them such as what we do for our model (Pre-trained Language Model). It notes that stance detection is not a pure binary classification problem, because detecting the underlying issue is required for identifying the polarity of opinion against it. Aside from the above notes, BiCoEn is designed for detecting the stance of tweets for one unseen single target (issue), however, in ProCon the size of input argument is much longer than a tweet (166 compared to 20 words) and belongs to a diverse number of issues. We set the maximum length of an input to be 20 words for IE and BiConEn, as recommended by the authors of [1]. However, we find that by increasing this threshold, accuracy decreases. The reason is that the sequence length of both LSTMs must be equal, because the initial weights of argument-LSTM are the output of issue-LSTM. When we increase the maximum length, issue-LSTM takes no new information but padding indices (the average length of argument sequence is much greater than average length of issue sequence, $166 \gg 3$).
Ultimately, our model achieves an accuracy increase of more than 5% compared to BoW+RF. It indicates that leveraging the context information along with LM Fine-Tuning helps the model identify the issue and the stance against it more accurately. We provide more analysis in the following sections.

6.1 Effect of Max-pooling

The fusion layer merges the information from previous layers for prediction. To understand what the model learns in this layer we plot the word scores in the

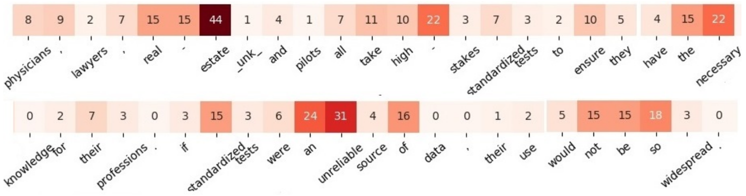

Fig. 2. Heatmap of max-pooling matrix of one argument. The underlying question: *"Is the Use of Standardized Tests Improving Education in America?"*. Darker colors show larger scores.

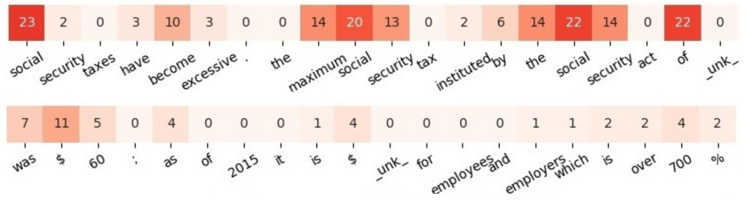

Fig. 3. Heatmap of max-pooling matrix of the first half of an argument (the second half scores are mostly zero). The underlying question: *"Should Social Security Be Privatized?"*. Darker colors show larger scores.

max-pooling matrix of an argument. We define the score of word w at time t, to be the index frequency of the embedding vector of w in pooling operation. The larger the score, the more important that word is to the model, because more embedding dimensions of that word appear in the max-pooled matrix (same word in different time steps may have different scores). Figure 2 and 3 show the heatmaps of a short and the first half of a longer argument respectively that are correctly classified. We cannot provide more plots due to space constraints. However, we find that the words at the beginning of long documents are more informative (Figure 3). One reason is that the first sentence of long arguments is *usually* the topic sentence that conveys the stance. Moreover, for shorter arguments, the model finds the information across all parts of the argument almost evenly.

6.2 Effect of pre-trained LM

To assess the impact of pre-trained LM, we examine our model without utilizing the pre-trained LM. We do not fine-tune the LM units over the training data, too. The experiment are represented in Table 5. The dramatic drop in all metrics shows the effect of the ablated techniques. Pre-training helps generalization and prevents our model from overfitting the relatively small training data.

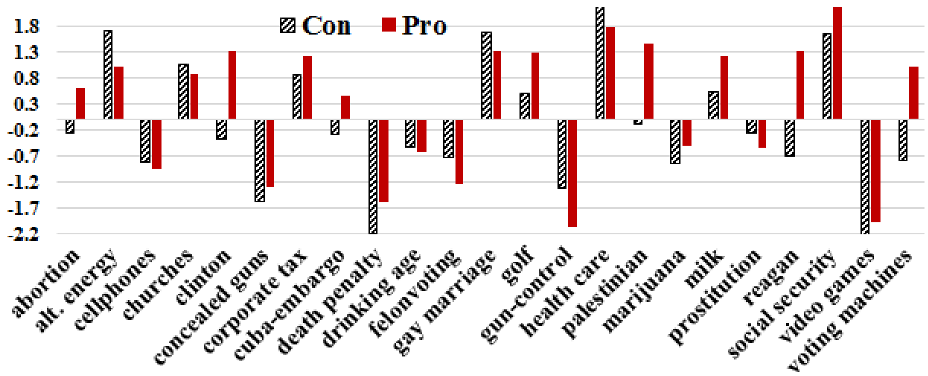

Fig. 4. Average sentiment score per issue per class.

Table 5. Effect of LM Fine-Tuning

Method	LM-FT	Pro			Con			**Macro-F1**	**Acc**
		P	R	F1	P	R	F1		
Our model	no	53.2	68.6	59.9	56.3	40.2	46.9	53.4	54.3
Our model	yes	65.9	82.6	73.3	77	57.7	65.9	69.6	70.1

6.3 Sentiment Analysis

How does sentiment relate to stance? Are pro-opinions often positive while cons are negative? To answer this question and find the relation between the stance and sentiment we define the sentiment score s_d of document d as $s_d = \sum_{s \in d} s_v$ where s_v is the VADER sentiment score of sentence s [6]. We compare the average sentiment score of the 23 issues from training set arguments between two classes (Figure 4). According to the plot, in some cases such as *abortion* and *voting machines* the score of pro is positive while con has negative overall score, indicating that proponents and opponents have different sentiments in their arguments about the issue. For some other cases, such as *health care*, both classes have a positive sentiment score. We identify as a key reason the concept of *'the right to health'*, has a positive sentiment. That makes opponents use this concept and its synonyms frequently making their arguments statistically positive. For "churches" where the underlying question is "should churches remain tax-exempt?" con has a larger positive score than pro. We find that some supporters (pro class) bring negative justifications by predicting the unsatisfactory situation after withdrawing the tax-exempt for churches. This unsatisfactory situation is explained while having negative sentiment.

7 Conclusion

We propose a general model for stance detection of arguments. Unlike most models, our documents are long (with the average size of 166 words) and come from a large number of different domains. Experiments show promising results compared to the baselines. We also find our proposed model relies on the beginning of long arguments for stance detection. And depending on the discussed issue, sentiment of an argument varies in pro or con class. Namely, pro-arguments express negative while con-argument have positive sentiment.

8 Acknowledgement

This work is supported in part by grants NSF 1838145, NSF 1527364, and NSF 1838147. We also thank anonymous reviewers for their helpful feedback.

References

1. Augenstein, I., Rocktäschel, T., Vlachos, A., Bontcheva, K.: Stance detection with bidirectional conditional encoding. arXiv preprint arXiv:1606.05464 (2016)
2. Bar-Haim, R., Bhattacharya, I., Dinuzzo, F., Saha, A., Slonim, N.: Stance classification of context-dependent claims. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. vol. 1, pp. 251–261 (2017)
3. Chen, W.F., Ku, L.W.: Utcnn: a deep learning model of stance classificationon on social media text. arXiv preprint arXiv:1611.03599 (2016)
4. Du, J., Xu, R., He, Y., Gui, L.: Stance classification with target-specific neural attention networks. International Joint Conferences on Artificial Intelligence (2017)
5. Ebrahimi, J., Dou, D., Lowd, D.: A joint sentiment-target-stance model for stance classification in tweets. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers. pp. 2656–2665 (2016)
6. Gilbert, C.H.E.: Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf (2014)
7. Hasan, K.S., Ng, V.: 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 (EMNLP). pp. 751–762 (2014)
8. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 328–339 (2018)
9. Levy, R., Bilu, Y., Hershcovich, D., Aharoni, E., Slonim, N.: Context dependent claim detection. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 1489–1500 (2014)
10. Merity, S., Keskar, N.S., Socher, R.: Regularizing and optimizing lstm language models. arXiv preprint arXiv:1708.02182 (2017)
11. Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., Cherry, C.: Semeval-2016 task 6: Detecting stance in tweets. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016). pp. 31–41 (2016)
12. Mohammad, S.M., Sobhani, P., Kiritchenko, S.: Stance and sentiment in tweets. ACM Transactions on Internet Technology (TOIT) **17**(3), 26 (2017)