NLPZZX at SemEval-2018 Task 1: Using Ensemble Method for Emotion and Sentiment Intensity Determination

Zhengxin Zhang, Qimin Zhou, Hao Wu
School of Information Science and Engineering, Yunnan University
Chenggong Campus, Kunming, P.R. China
{zzxynu,zqmynu}@gmail.com, haowu@ynu.edu.cn

Abstract

In this paper, we put forward a system that competed at SemEval-2018 Task 1: "Affect in Tweets". Our system uses a simple yet effective ensemble method which combines several neural network components. We participate in two subtasks for English tweets: EI-reg and V-reg. For two subtasks, different combinations of neural components are examined. For EI-reg, our system achieves an accuracy of 0.727 in Pearson Correlation Coefficient (all instances) and an accuracy of 0.555 in Pearson Correlation Coefficient (0.5-1). For V-reg, the achieved accuracy scores are respectively 0.835 and 0.670.

1 Introduction

Sentiment analysis is a research area in the field of natural language processing. It aims to detect the sentiment expressed by the author of some form of textual data and many deep learning approaches have been successfully exploited (Cambria, 2016). The goal of SemEval-2018 Task 1 "Affect in Tweets" is to automatically determine the intensity of emotions and intensity of sentiment of the tweeters from their tweets (Mohammad et al., 2018). All tweets fall into three languages: *English*, *Arabic* and *Spanish*. We participate in two subtasks for English tweets: EI-reg and V-reg. For EI-reg, all English tweets are separated into four emotions, anger, fear, joy and sadness. Every emotion has train, dev and test datasets. This subtask determines the intensity which is a real-valued score between 0 and 1 of emotion that represents the mental state of the tweeter. The instances with higher scores correspond to a greater degree of emotion than instances with lower scores. For V-reg, all English tweets are divided into three datasets: train, dev and test datasets. It determines the intensity of sentiment or valence that best represents the mental state of the tweeter a real-valued score between 0 and 1. The instances with higher scores correspond to a greater degree of positive sentiment than instances with lower scores. Both the two subtasks are regression tasks.

For these two subtasks, we have adopted separate ensemble method with existing neural network components (Brueckner and Schulter, 2014; Kim, 2014; Li and Qian, 2016; Yang et al., 2017) (see Figure 1). We use BiLSTM-CNN component, BiLSTM-Attention component and Deep BiLSTM-Attention component with different embeddings for simple ensemble. In these subtasks, our final model is just an average of scores provided by what we select from these single neural network components. Every emotion or valence employs different ensemble method, so there are several distinct ensemble methods in the two subtasks. Experimental results show that our proposed ensemble methods are simple yet effective.

The remainder of the paper is structured as follows. We provide details of the proposed ensemble method in Section 2. We present the experimental result of proposed methods in Section 3. Finally, a conclusion is drawn in section 4.

2 Methodology

We propose an simple ensemble method of different neural network components. We mainly introduce the implementation details of these components, including raw tweets preprocessing, lexicon features and embedding resources we use in these components, the architecture of these components and the best parameters of different single components. The parameters that can maximize the Pearson Correlation Coefficient between the predicted values and real values are chosen to be the best parameters.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 116–122
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

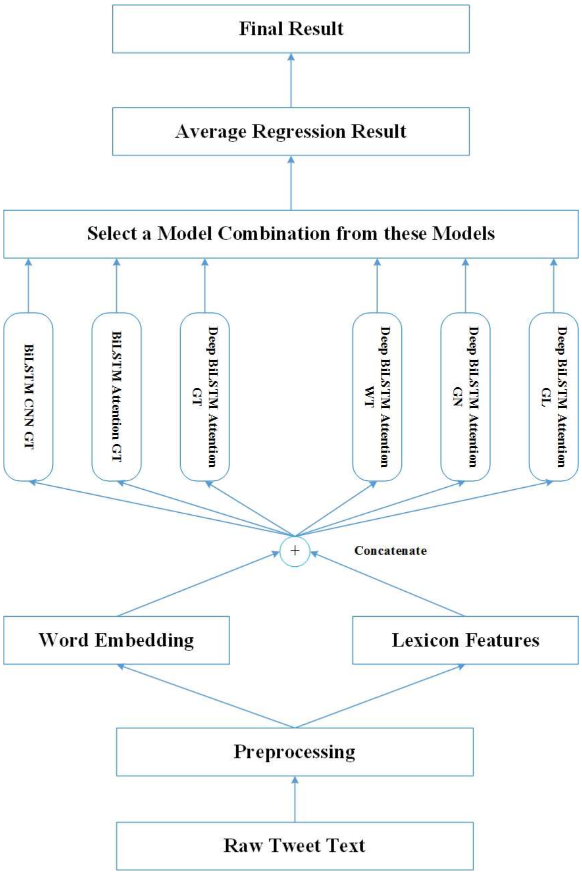

Figure 1: The architecture of our system.

2.1 Data Preprocessing

In general, tweet are not always syntactically well-structured and the language used does not always strictly adhere to grammatical rules (Barbosa and Feng, 2010). So we need to preprocess raw tweets before feature extraction. Firstly, we perform a few preprocessing steps, such as remove # and retain the word itself, remove stop words with nltk.corpus. Then the tweets are transformed into lowercase. Finally, we utilize TweetTokenizer[1] to process the tweets.

2.2 Feature Extraction

Each tweet is represented as a concatenation of two different feature vectors, one is lexicon features and another is word embedding. In our system, each tweet is divided into words, every word is represented as a $d + m$ dimension vector and thus each tweet is represented as $l(d + m)$ matrix, where d is the dimension of word embedding and m is the dimension of lexicon features. Suppose each tweet has the same length, so l is the length of tweet. We utilize a variety of resources for feature extraction as follows:

1. AFINN: Calculating positive and negative sentiment scores from the lexicon (Nielsen, 2011).

2. NRC Affect Intensity Lexicon: The NRC Affect Intensity Lexicon is a list of English words and their associations with four basic emotions (anger, fear, sadness, joy) (Mohammad, 2017).

3. NRC Emotion Lexicon: The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive) (Mohammad and Turney, 2010).

4. NRC Hashtag Emotion Lexicon: Association of words with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) generated automatically from tweets with emotion-word hashtags (Mohammad, 2012).

5. NRC Emoticon Lexicon: Association of words with positive (negative) sentiment generated automatically from tweets with emoticons (Kiritchenko et al., 2014; Mohammad et al., 2013; Zhu et al., 2014).

6. NRC Emoticon Affirmative Context Lexicon and NRC Emoticon Negated Context Lexicon: Association of words with positive (negative) sentiment in affirmative or negated contexts generated automatically from tweets with emoticons (Kiritchenko et al., 2014; Mohammad et al., 2013; Zhu et al., 2014).

7. NRC Hashtag Affirmative Context Sentiment Lexicon and NRC Hashtag Negated Context Sentiment Lexicon: Association of words with positive (negative) sentiment in affirmative or negated contexts generated automatically from tweets with sentiment-word hashtags (Kiritchenko et al., 2014; Mohammad et al., 2013; Zhu et al., 2014).

8. NRC Hashtag Sentiment Lexicon: Association of words with positive (negative) sentiment generated automatically from tweets

[1] http://www.nltk.org/

with sentiment-word hashtags (Kiritchenko et al., 2014; Mohammad et al., 2013; Zhu et al., 2014).

9. Emoji: This is a manual classification of the dictionary, in which each emoji has a corresponding polarity value.

10. Sentiwordnet: Sentiwordnet is a lexical resource explicitly devised for supporting sentiment classification and opinion mining applications (Baccianella et al., 2010), through the wordnet entry in the emotional classification, and marked each entry belongs to the positive and negative categories weight size.

2.3 Neural Networks

2.3.1 Embeddings

The final model combines three neural network components as *BiLSTM-CNN*, *BiLSTM-Attention*, and *Deep BiLSTM-Attention*. Towards BiLSTM-CNN and BiLSTM-Attention, we use *glove.twitter.27B.200d* which contains pre-trained word vectors with Glove algorithm (Pennington et al., 2014). For Deep BiLSTM-Attention, different pre-trained word vectors are used, such as *word2vec-twitter-model*, *GoogleNews-vectors-negative300*, *glove.twitter.27B.200d and glove.840B.300d*.

1. word2vec-twitter-model [2]: word2vec model (Mikolov et al., 2013) is a NLP tool launched by Google in 2013. It features the quantification of all words so that words can be quantified to measure the relationship between them. word2vec-twitter-model is trained on tweets and the embedding dimension used in our system is 400.

2. GoogleNews-vectors-negative300 [3]: GoogleNews vectors is trained on Google News corpus. It resembles word2vec-twitter-model and the embedding dimension is 300.

3. glove.840B.300d [4]: Glove is an unsupervised learning algorithm for obtaining vector representations for words. Training is conducted on aggregated co-occurrences of words from a global corpus, and the resulting representations showcase interesting linear substructures of the word vector space. The embedding dimension used in our system is 300.

4. glove.twitter.27B.200d [4]: This word embedding is trained on 2 billion tweets from twitter. It is similar to glove.840B.300d, but the embedding dimension is 200.

2.3.2 Bidirectional LSTM with CNN

The BiLSTM with CNN first transform tweets into text matrices, the BiLSTM is applied to these matrices to build new text matrices, CNN is applied to the output of the BiLSTM to obtain text vectors for the prediction of emotional intensity. The BiLSTM with CNN achieves a rather good result on the task of emotional analysis (He et al., 2017). so we choose it for our task.

Model Architecture: Embedding vectors are fed into a BiLSTM network followed by a CNN layer. The CNN layer consists of one dimensional convolutional layer and pooling layer where the number of filters is 256, the window size of the filter is 3, and the *activation function* is Relu. The input and output shape of convolutional layer are both 3D tensor. The output of the CNN layer is flattened after max-pooling operation. After the Flatten layer, two dense layers are stacked and the *activation functions* are respectively configured as Relu and Sigmoid. Also dropout (Srivastava et al., 2014) is utilized to avoid potential overfitting, it is used between two dense layers. The reason why we select Relu is to prevent the vanishing gradient problem and accelerate the calculation. Since the task is a regression problem, we put a dense projection with sigmoid activation to obtain an intensity value between 0 and 1.

Model Training: The network parameters are learned by minimizing the *mean squared error* (MSE) between the real and predicted values of emotion intensity or valence intensity. We optimize this loss function via *Adam* that is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments (Kingma and Ba, 2014). Batch size and training *epochs* may be different for different emotions and valence. To avoid overfitting issues, we use dropout in this model. Finally, we apply these three parameters for system tuning. In addition, we try various optimization algorithms with the same param-

[2] http://www.spark.tc/building-a-word2vec-model-with-twitter-data/

[3] https://github.com/mmihaltz/word2vec-GoogleNews-vectors

[4] https://nlp.stanford.edu/projects/glove/

EI-reg	Anger			Fear			Joy			Sadness		
	BS	Epochs	Dp	BS	Epochs	Dp	BS	Epochs	Dp	BS	Epochs	Dp
BiLSTM CNN+GT	16	6	0.5	32	2	0.5	8	4	0.5	32	5	0.5
BiLSTM Attention+GT	32	3	0.5	32	3	0.5	8	7	0.5	32	4	0.5
Deep BiLSTM Attention	32	2	0.3	8	7	0.3	16	9	0.1	16	5	0.6

Table 1: The best parameters of EI-reg.

V-reg	Valence		
	BS	Epochs	Dp
BiLSTM CNN+GT	8	5	0.5
BiLSTM Attention+GT	8	10	0.6
Deep BiLSTM Attention+WT	16	8	0.5
Deep BiLSTM Attention+GN	32	10	0.5
Deep BiLSTM Attention+GL	8	5	0.2
Deep BiLSTM Attention+GT	16	8	0.2

Table 2: The best parameters of V-reg.

eters, such as *SGD*, *RMSprop*, *Adagrad*, *Adam* and *Adamax*, and find that Adam works best. So we fix the optimization algorithm with Adam (Kingma and Ba, 2014) and tune the parameters, the best configurations for EI-reg and V-reg are respectively given in Tables 2.3.1 and 2.3.1, where *BS* is batch size, *Dp* is dropout.

2.3.3 Bidirectional LSTM with Attention

Bidirectional LSTM with Attention achieves a good result on the SemEval-2017 Task 4 “Sentiment Analysis in Twitter” (Baziotis et al., 2017), so we exploit *Bidirectional LSTM with Attention model* and *Deep Bidirectional LSTM with Attention model* for our tasks.

Model Architecture: For Bidirectional LSTM with attention model, embedding vectors are fed into a BiLSTM network followed by an attention layer (Yang et al., 2017). Not all words contribute equally to the expression of sentiment in a tweet, so we use an attention layer to find the importance of each word in tweet. After the attention layer, it is consistent with Bidirectional LSTM with CNN model. The difference between the Bidirectional LSTM with attention model and its deep version is that, we use two BiLSTM layers followed by an attention layer in the deep version.

Model Training: We use the same method to learn the network parameters. In EI-reg, we use the same batch size, training *epochs* and dropout to train the Deep BiLSTM Attention model with different pre-training word embeddings in every emotion, but in V-reg, batch size, training *epochs* and dropout are different in Deep BiLSTM Attention model with different pre-training word embeddings. In these models, we also use dropout. The best parameters of EI-reg for these models are given in Table 2.3.1 and V-reg’s best parameters are given in Table 2.3.1.

2.4 Ensemble Methods

Currently, ensembling is a widely used strategy which combines multiple single components to improve overall performance, there are many ensemble methods that have been proposed, such as, Voting, Blending, Bagging, Boosting, etc [5]. In this system, due to time constraint, we choose a simple average of the scores provided by different components, as each single component can predict emotional intensity or valence intensity. It can be defined as

$$Prediction_{intensity} = \sum_{i=1}^{n} \frac{model_i}{n} \qquad (1)$$

where n is the number of neural components. Model$_i$ represents the prediction results of i-th component. Suppose three components are exploited to predict the intensity of anger, and three prediction values of a same tweet 0.76, 0.72 and 0.7 are suggested, then the final result of this tweet will be $(0.76 + 0.72 + 0.74)/3 = 0.74$.

3 Experiments

Dataset	train	dev	test	sum
anger	1,701	388	17,939	20,028
fear	2,252	389	17,923	20,564
joy	1,616	290	18,042	19,948
sadness	1,533	397	17,912	19,842
valence	1,181	449	17,874	19,504

Table 3: Statistics of the datasets.

For experiments, we use five datasets from two different subtasks, These datasets, “EI-reg-En-anger (anger)”, “EI-reg-En-joy (joy)”, “EI-reg-En-fear (fear)”, “EI-reg-En-sadness (sadness)” and “2018-Valence-reg-En (valence)” are downloaded from SemEval-2018 Task 1 “Affect in Tweets” [6]. As for the EI-reg task dataset format, each tweet

[5] https://mlwave.com/kaggle-ensembling-guide/

[6] https://competitions.codalab.org/competitions/17751

EI-reg	Average		Anger		Fear		Joy		Sadness	
	All	**0.5-1**	**All**	**0.5-1**	**All**	**0.5-1**	**All**	**0.5-1**	**All**	**0.5-1**
Baseline	0.520	0.396	0.526	0.455	0.525	0.302	0.575	0.476	0.453	0.350
BiLSTM CNN+GT	-	-	-	-	0.691	0.508	0.701	0.512	0.694	0.507
BiLSTM Attention+GT	-	-	0.701	0.583	0.715	0.506	0.711	0.513	0.720	0.557
Deep BiLSTM Attention+WT	-	-	0.697	0.582	0.709	0.507	0.728	0.503	0.704	0.541
Deep BiLSTM Attention+GN	-	-	0.681	0.557	-	-	-	-	0.698	0.535
Deep BiLSTM Attention+GL	-	-	-	-	-	-	-	-	-	-
Deep BiLSTM Attention+GT	-	-	-	-	-	-	-	-	0.717	0.551
Ensemble	**0.727**	**0.555**	**0.716**	**0.607**	**0.726**	**0.519**	**0.736**	**0.529**	**0.729**	**0.565**

Table 4: Performance comparisons of models in different emotions, where the best values are marked in **bold**.

consists of the id, the tweet, the emotion of the tweet, the emotion intensity and for the V-reg task, each tweet consists of the id, the tweet, the sentiment of the tweet and the sentiment intensity. All datasets have been divided into train set, dev set and test set. Test set's gold labels are given only after the evaluation period. Statistics of the datasets are shown in Table 3.

To measure the performance of selected methods, two submetrics of Pearson Correlation Coefficient (PCC) are used. PCC (all instances) is Pearson correlation for a subset of test data that includes all tweets. The value varies between -1 and 1. PCC (0.5-1) is the Pearson correlation for a subset of test data that includes only those tweets with intensity score greater or equal to 0.5. For both metrics, a larger value indicate a better prediction accuracy.

For each dataset, we use dev set to select our ensemble methods. Firstly we run these six components on all dev datasets. Then, combine these results of different components, different combinations of components lead to different results on dev set. Finally, we select the combination with a higher score for testing.

Our system is implemented on Keras with a Tensorflow backend [7]. We present the result of PCC (all instances) and PCC (0.5-1) for each emotion and valence on the test data, shown in Tables 3 and 3. For simplicity, we denote *WT*, *GN*, *GL* and *GT* for the word vectors of *word2vec-twitter-model*, *GoogleNews-vectors-negative300*, *glove.840B.300d* and *glove.twitter.27B.200d*. We compare the results of our single components, official baseline and our ensemble system. Every emotion and valence adopts different ensemble methods, the symbol **'-'** means that the component is not used in the ensemble method in this emotion or valence. For example, we only use *BiLSTM Attention+GT*, *Deep BiLSTM Attention+WT* and *Deep BiLSTM Attention+GN* these three components for ensemble on anger dataset. The reason why we don't use all the six components for ensemble is that ensemble does not always have a good effect, a same component can have different effects on different datasets, either good or bad. The official result for EI-reg, our average PCC reaches 0.727 in all instances and 0.555 in 0.5-1 (both ranked 10 out of 48 participants). For V-reg, the result is 0.835 in all instances (ranked 7 out of 38) and 0.670 in 0.5-1 (ranked 6 out of 38). The average result of baseline for EI-reg is 0.520 and 0.396, for V-reg, the result is 0.585 and 0.449. These results demonstrate that the ensemble approach achieves important improvement in performance across all the emotions and valence, and gains the best performance for Anger.

V-reg	Valence	
	All	**0.5-1**
Baseline	0.585	0.449
BiLSTM CNN+GT	-	-
BiLSTM Attention+GT	-	-
Deep BiLSTM Attention+WT	0.825	0.665
Deep BiLSTM Attention+GN	0.820	0.640
Deep BiLSTM Attention+GL	0.822	0.648
Deep BiLSTM Attention+GT	0.825	0.659
Ensemble	**0.835**	**0.670**

Table 5: Performance comparisons of models in valence, where the best values are marked in **bold**.

4 Conclusions and Future Works

We have proposed a simple yet effective ensemble method which integrates various neural components to perform the sentiment or emotion analysis for the tweet. Experimental results reflect that our method is effective in the prediction tasks of emotional intensity and sentimental intensity. Some other useful findings can be drawn from the experimental results: a) The model of integration for each emotion is different; b) As for lexicon features and word embedding, it is important for emotion or sentiment analysis; c) ensemble is not al-

[7] https://keras.io/

ways valid. Also, we have tried data augmentation considering insufficient training data, however the effect is not a good.

As for future works, although our ensemble method has achieved good results, we would want to examine the multi-task deep learning approach on these tasks, by which it would predict the different emotional intensity at the same time, and improve the generalization effect of the prediction model.

Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (61562090) and the Graduate Research Innovation Fund Project of Yunnan University (YDY17113).

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *International Conference on Language Resources and Evaluation, Lrec 2010, 17-23 May 2010, Valletta, Malta*, pages 83–90.

Luciano Barbosa and Junlan Feng. 2010. Robust sentiment detection on twitter from biased and noisy data. In *International Conference on Computational Linguistics: Posters*, pages 36–44.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 747–754.

Raymond Brueckner and Bjorn Schulter. 2014. Social signal classification using deep blstm recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4823–4827.

Erik Cambria. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, 31(2):102–107.

Yuanye He, Liang-Chih Yu, K. Robert Lai, and Weiyi Liu. 2017. YZU-NLP at emoint-2017: Determining emotion intensity using a bi-directional LSTM-CNN model. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 238–242.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Dan Li and Jiang Qian. 2016. Text sentiment analysis based on long short-term memory. In *IEEE International Conference on Computer Communication and the Internet*, pages 471–475.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 321–327.

Saif M Mohammad. 2012. # emotional tweets. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 246–255. Association for Computational Linguistics.

Saif M Mohammad. 2017. Word affect intensities. *arXiv preprint arXiv:1704.08798*.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Saif M. Mohammad and Peter D. Turney. 2010. Emotions evoked by common words and phrases: using mechanical turk to create an emotion lexicon. In *NAACL Hlt 2010 Workshop on Computational Approaches To Analysis and Generation of Emotion in Text*, pages 26–34.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. In *Workshop on'Making Sense of Microposts: Big things come in small packages*, pages 93–98.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2017. Hierarchical attention networks for document classification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Xiaodan Zhu, Svetlana Kiritchenko, and Saif Mohammad. 2014. Nrc-canada-2014: Recent improvements in the sentiment analysis of tweets. In *International Workshop on Semantic Evaluation*, pages 443–447.