THU_NGN at SemEval-2018 Task 1: Fine-grained Tweet Sentiment Intensity Analysis with Attention CNN-LSTM

Chuhan Wu[1], Fangzhao Wu[2], Junxin Liu[1],Zhigang Yuan[1],
Sixing Wu[1] and Yongfeng Huang[1]
[1]Tsinghua National Laboratory for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University Beijing 100084, China
[2]Microsoft Research Asia
{wuch15,wu-sx15,ljx16,yuanzg14,yfhuang}@mails.tsinghua.edu.cn
wufangzhao@gmail.com

Abstract

Traditional sentiment analysis approaches mainly focus on classifying the sentiment polarities or emotion categories of texts. However, they can't exploit the sentiment intensity information. Therefore, the SemEval-2018 Task 1 is aimed to automatically determine the intensity of emotions or sentiment of tweets to mine fine-grained sentiment information. In order to address this task, we propose a system based on an attention CNN-LSTM model. In our model, LSTM is used to extract the long-term contextual information from texts. We apply attention techniques to selecting this information. A CNN layer with different kernel sizes is used to extract local features. The dense layers take the pooled CNN feature maps and predict the intensity scores. Our system achieves an average Pearson correlation score of 0.722 (ranked 12/48) in the emotion intensity regression task, and 0.810 in the valence regression task (ranked 15/38). It indicates that our system can be further extended.

1 Introduction

Detecting the intensity of sentiment is an important task for fine-grained sentiment analysis (Kiritchenko et al., 2016; Mohammad and Bravo-Marquez, 2017). Intensity refers to the degree or amount of an emotion or degree of sentiment. For example, we can express our emotion by "very happy" or "a little angry". The intensity can be analysis in multiple categories (i.e. low, moderate and high) or real-valued. Identifying the intensity information of sentiment has potential to applications such as electronic business, social computing and public health (Wilson, 2008).

Twitter is a social platform which contains rich textual content. There have been many approaches to twitter sentiment analysis (Khan et al., 2015; Severyn and Moschitti, 2015; Philander et al., 2016). However, twitter sentiment analysis is challenging because tweets usually contain nonstandard languages, including emoticons, emojis, creatively spelled words, and hash tags (Mohammad and Bravo-Marquez, 2017). In order to improve the collective techniques on tweet sentiment intensity analysis, the SemEval-2018 Task 1 is aimed to identify the categorical and real-valued intensity of emotions or sentiment for English, Arabic, and Spanish (Mohammad et al., 2018).

Existing approaches to analysis the intensity of emotions or sentiment are mainly based on lexicons and supervised learning. Lexicon-based methods usually rely on lexicons to assign the intensity scores of affective words in texts (Mohammad and Bravo-Marquez, 2017). However, these method can't utilize the contextual information from texts. Supervised methods are mainly based on SVR (Madisetty and Desarkar, 2017), linear regression (John and Vechtomova, 2017) and neural networks (Goel et al., 2017; Köper et al., 2017). Usually neural network-based methods outperform SVR and linear regression-based methods siginificantly. Motivated by the successful applications of neural models in this task, we propose a system using a CNN-LSTM model with attention mechanism. Firstly, a tweet will be converted into a sequence of dense vectors by an embedding layer. Next, we use a Bi-LSTM layer to extract contextual information from them. The sequential features will be selected by an attention layer. Then we apply a CNN with different kernel sizes to extracting different local information. Thus, our model can exploit both local and long-term information by combining CNN and LSTM. Finally, two dense layers are used to predict the intensity scores. The system performance quantified by an average Pearson correlation score is 0.722 in the emotion intensity regression task (EI-reg) and 0.810 in the valence regression task (V-

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 186–192
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

reg). Our model outperforms several baseline neural networks, which proves that our model can identify the intensity of emotions and sentiment effectively.

2 Related Work

Sentiment analysis in social media such as Twitter is an important task for opinion mining (Severyn and Moschitti, 2015). Traditional Twitter sentiment analysis methods mainly focus on identifying the polarities (Da Silva et al., 2014; dos Santos and Gatti, 2014) or emotion categories (Dini and Bittar, 2016) of tweets. However, it's a difficult task to analysis the noisy tweets. They usually contain various nonstandard languages including emoticons, emojis, creatively spelled words and hash tags. In addition, these languages usually contain rich sentiment information. In order to capture such information, several lexicon-based methods are proposed. Nielsen et al. (2011) proposed to use a dictionary to incorporate emoticon information into tweet analysis models. Mohammad et al. proposed to use hash tags to identify emotion categories of tweets (2015). These lexicon-based methods are free from manual annotation, but they rely on the emotion lexicons and can't mine high-level contextual information from tweets. Supervised methods such as neural networks are also applied to tweet sentiment analysis. For example, Dos et al. (2014) propose to classify tweets using a deep convolutional neural network. Approaches based on deep neural networks need sufficient samples to train, but they usually outperforms lexicon-based methods in these tasks.

However, these approaches usually ignore the intensity of emotions and sentiment, which provides important information for fine-grained sentiment analysis. Therefore, in order to capture such information, Mohammad et al. proposed to identify the emotion and sentiment intensity (valence) of texts (2016). Different approaches have been proposed to detect the tweet emotion intensity in the EmoInt-2017 shared task (Mohammad and Bravo-Marquez, 2017). For example, Madisetty et al. (2017) proposed an ensemble model based on SVR. Goel et al. (2017) and Koper et al. (2017) applied CNN-LSTM architecture to this task. These systems reached the top ranks in the EmoInt shared task.

Motivated by the successful application of CNN-LSTM model (Zhou et al., 2015; Chen et al., 2016) and the attention mechanism for text classification (Yin et al., 2015), we propose a system using attention-based CNN-LSTM model to address this task. In our model, we first use LSTM to extract sequential information, and select features via attention layer. Then we combine CNN with different kernel sizes to learn local information. Finally the dense layers are used to predict the intensity scores. In addition, several features are incorporated into our model. The evaluation results show that our system outperform several baseline neural networks and can be further extended.

3 Attention CNN-LSTM Model

Our network architecture is shown in Figure 1. We will explain the detailed information of our system in the following subsections.

3.1 Network Architecture

As shown in Figure 1, an embedding layer is used to provide word embedding and one-hot encoded part-of-speech (POS) tags of the input tweets. The Bi-LSTM layer takes the concatenated word embedding and POS tags as input, and output each hidden states. Let h_i be the output hidden state at time step i. Then its attention weight α_i can be formulated as follows:

$$\begin{aligned} m_i &= tanh(h_i), \\ \hat{\alpha_i} &= w_i m_i + b_i, \\ \alpha_i &= \frac{exp(\hat{\alpha_i})}{\sum_j exp(\hat{\alpha_j})}, \end{aligned} \tag{1}$$

where $w_i m_i + b_i$ denote a linear transformation of m_i. Therefore, the output representation r_i is given by:

$$r_i = \alpha_i h_i. \tag{2}$$

Based on such text representation, the sequence of features will be assigned with different attention weights. Thus, important information such as affective words can be identified more easily. The convolutional layer takes the text representation r_i as input. We use CNN with four different kernel sizes to learn local information with different contextual length. Based on this architecture, our model can combine both long-term and local information, which can help to identify sentiment information better. The output CNN feature maps are concatenated together, and will be squeezed by a global max pooling layer. They are concatenated with the lexicon features. We use two dense layers

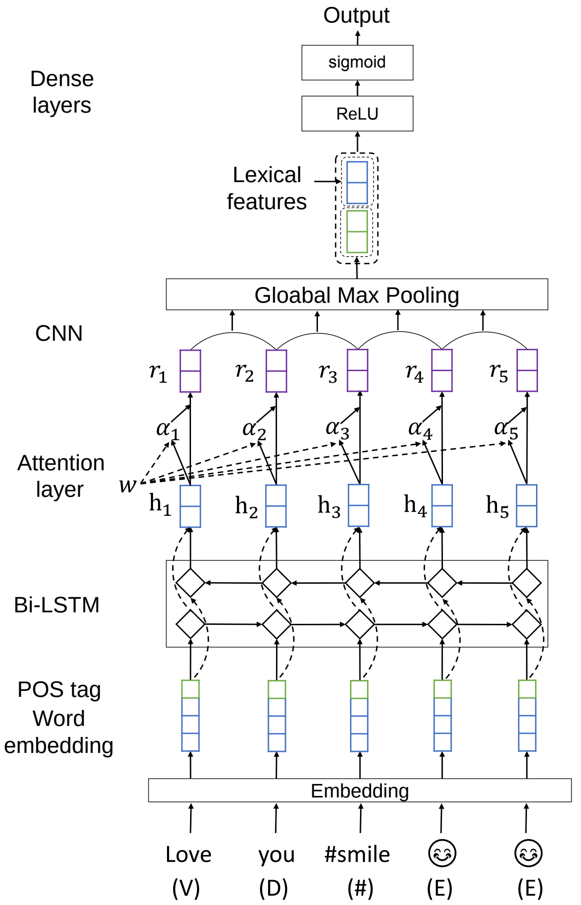

Figure 1: The architecture of our attention CNN-LSTM model.

with ReLU and sigmoid activation respectively to predict the final intensity score. In order to mitigate overfitting, we apply dropout technique at each layer to regularize our model.

3.2 Word Embedding

We use Word2Vec (Mikolov et al., 2013) as the vector representation of the words in tweets. We combine two kinds of word embeddings: The first embeddings are provided by Godin et al. (2015). They are trained on a corpus with 400 million tweets. The second embeddings are provided by Barbier et al. (2016). They are trained on 20 million geolocalized tweets. The dimensions of two embeddings are 400 and 300 respectively. We fine-tune the word embeddings during the network training.

3.3 Additional Features

We incorporate POS tags and lexicon features into our model. POS tags usually contain rich semantic information. For example, sentiment intensity can be expressed by adjectives like “very” and “slight”. POS tags can help the neural model to identify such words. We use the Ark-Tweet-NLP[1] tool to obtain the POS tags of tweets (Owoputi et al., 2013). The POS tag feature of each word is concatenated with the word embedding.

Usually affective words in tweets such as specific hashtags express sentiment explicitly. Therefore, incorporating lexicon information can help our model to predict intensity more accurately. We use the AffectiveTweets[2] (Mohammad and Bravo-Marquez, 2017) package in Weka[3] to obtain the lexicon features of tweets. We use the TweetToLexiconFeatureVector (Bravo-Marquez et al., 2014), TweetToSentiStrengthFeatureVector (Thelwall et al., 2012) and TweetToInputLexiconFeatureVector filters in AffectiveTweets. In our experiment, the lexicon features are 49-dim. These lexicon features are concatenated with the pooled CNN feature maps.

3.4 Model Ensemble

We use an ensemble strategy to improve the model performance. Our model is trained for 10 times by using randomly selected dropout rate. Then the final predictions on the test set are given by the average of all model predictions. In this way, the random error of our system can be reduced.

4 Experiment

4.1 Preprocess

In order to process the noisy tweet texts, we use tweetokenize[4] for tokenizing, and use Ark-Tweet-NLP tool for POS tagging. In addition, we refine the texts and POS tags using several rules: 1) all URLs will be replaced with the word “URL”, and their POS tags will be set to “URL”; 2) all @users will be replaced with “USERNAME”, and their POS tags will be set to @; 3) POS tags of hashtags are set to “#”; 4) POS tags of emojis and emoticons are set to “E”.

[1] http://www.cs.cmu.edu/ ark/TweetNLP
[2] https://github.com/felipebravom/AffectiveTweets
[3] https://www.cs.waikato.ac.nz/ml/weka
[4] https://github.com/jaredks/tweetokenize

4.2 Experiment Settings

The details of English datasets[5] we use is shown in Table 1. The intensity in both task is annotated between 0 and 1. In the EI-reg task, the Pearson correlation scores across all four emotions will be averaged as the final score. In the V-reg task, the correlation score for valence is used as the competition metric.

Task	EI-reg				V-reg
Category	anger	fear	joy	sadness	valence
#train	1,701	2,252	1,616	1,533	1,174
#dev	388	389	290	397	449
#test	1,002	986	1,105	975	937

Table 1: Detailed statistics of the English datasets in our experiment

In our network, the dimension of word embeddings is $400+300$. The hidden states of Bi-LSTM are 2×300-dim. The kernel sizes of CNN are 3, 5, 7 and 9 respectively. The number of feature maps are 4×200. The dimension of the first dense layer is set to 200. The padding length of tweets is set to 50. The dropout rate is a random number between 0.1 and 0.3. The loss function we use is MAE, and the batch size is set to 8. We combine the training and development sets in our experiment. We use 90% for training and reserve 10% for cross validation. In our official submissions, we use the full training and development sets to train models.

4.3 Evaluation Results

We compare the performance of our model and several baselines. The models to be compared include: 1) CNN, using CNN and dense layers. 2) LSTM, using LSTM and dense layers. 3) CNN+LSTM, combing CNN with LSTM to predict. 4) CNN+LSTM+att, adding attention mechanism to CNN-LSTM model. 5) CNN+LSTM+att+ensemble, using ensemble strategy in the attention-based CNN-LSTM model. The results in the EI-reg and V-reg tasks are shown in Table 2. In comparison, we also present the cross validation results. Our system reaches average Pearson correlation score of 0.722 in the EI-reg task and 0.810 in the V-reg task. The results indicate that our CNN-LSTM model outperforms the CNN and LSTM baselines. It proves that CNN-LSTM model can combine the long-term information and local information in texts. The attention mechanism can also improve the model performance. Since the attention layer can select important information, our model can focus on important words in texts (e.g. affective words) to predict the intensity of emotions and sentiment more accurately. Although our system still needs to be improved compared with the top systems, our model outperforms the common baseline models, which validates the effectiveness of our model.

4.4 Influence of Pre-trained Word Embedding

We compare the performance using different pre-trained embeddings in the EI-reg task. The results are shown in Table 3. The results show that the pre-trained embeddings are important, and combining different word embedding can improve the model performance. It may be because the combination of embedding can cover more out-of-vocabulary words and provide rich semantic information.

4.5 Influence of Additional Features

The influence of the POS tag features and lexicon features is shown in Table 4. The results show that POS tags can improve the model performance significantly. Affective words, emojis and hashtags usually contain rich sentiment information. POS tags can be used to identify such words. Therefore, incorporating the POS information into our neural model can help to identify these words in tweets better. The lexicon features can also improve our model. The lexicon features are obtained by the sentiment words in tweets. Thus, incorporating these features into neural networks can improve the performance of our system.

4.6 Analysis of Inappropriate Biases

In the EI-reg and V-reg tasks, an automatically generated mystery set is used for testing the inappropriate biases in NLP systems, such as gender and race (i.e. African American and European American names). For example, the pairs of sentences “She is happy.” and “He is happy.”; “Jamel feels angry.” and “Harry feels angry.” should be assigned wit the same intensity by an unbiased NLP system. The score differences are calculated for such sentence pairs. The average score difference, the p-value, and whether the score differences are statistically significant are shown in Ta-

[5] http://www.saifmohammad.com/WebDocs/AIT-2018/AIT2018-DATA

Model	EI-reg										V-reg	
	macro-avg		anger		fear		joy		sadness		valence	
	val	test	val	test	val	test	val	test	val	test	val	test
CNN	0.743	0.710	0.700	0.726	0.759	0.701	0.771	0.727	0.742	0.686	0.809	0.790
LSTM	0.741	0.706	0.701	0.720	0.751	0.694	0.766	0.726	0.746	0.683	0.802	0.785
CNN+LSTM	0.743	0.713	0.705	0.730	0.758	0.701	0.770	0.735	0.740	0.687	0.815	0.796
CNN+LSTM+att	0.749	0.718	0.706	0.731	0.760	0.706	0.774	0.739	0.756	0.695	0.828	0.801
CNN+LSTM+att+ensemble	**0.758**	**0.722**	**0.720**	**0.734**	**0.771**	**0.710**	**0.782**	**0.743**	**0.760**	**0.700**	**0.845**	**0.810**

Table 2: Evaluation and cross validation performance of our model ande baselines.

Embedding	avg	anger	fear	joy	sadness
w/o pre-trained	0.669	0.678	0.672	0.682	0.645
+emb1	0.717	0.728	0.706	0.737	0.695
+emb2	0.709	0.716	0.702	0.728	0.691
+emb1+emb2	**0.722**	**0.734**	**0.710**	**0.743**	**0.700**

Table 3: Influence of using different combinations of pre-trained word embeddings. The emb1 and emb2 denote the embeddings provided by Godin et al. (2015) and Barbieri et al. (2016) respectively.

Feature	avg	anger	fear	joy	sadness
None	0.704	0.715	0.698	0.722	0.679
+POS	0.715	0.729	0.705	0.737	0.690
+Lexicon	0.708	0.721	0.700	0.726	0.684
+POS+Lexicon	**0.722**	**0.734**	**0.710**	**0.743**	**0.700**

Table 4: Influence of POS tags and lexicon features.

ble 5. Although the average differences are small, but they are statistical significant in most tasks. Our system is based on word embedding, and we fine-tune the weights during the network training. Thus, our system will be influenced by the distribution of training data, which may lead to these biases.

Task	Gender			Race		
	Avg-D	p	Sig	Avg-D	p-value	Sig
Anger	-0.002	0.00003	√	0.002	0.01553	×
Fear	-0.023	0	√	0.023	0	√
Joy	0.02	0	√	-0.04	0	√
Sadness	-0.001	0.09654	×	0.011	0	√
Valence	0.001	0.00382	×	-0.021	0	√

Table 5: The average differences, p-value and statistical significance of predictions on the mystery set in each task. We denote them as Avg-D, p and Sig respectively.

4.7 Visualization of Attention Mechanism

Attention mechanism can encourage the neural model to focus on important words in texts. In order to prove its effectiveness of the attention layer, we present several examples in Table 6. The green color represents low attention, while red color represents high attention. We can see that the affective words (e.g. Happy) and hashtags (e.g. #funny) have high attention weights. It indicates that our attention-based model can capture important sentiment information to predict the intensity of tweets better.

5 Conclusion

Identifying the intensity of emotions or sentiment is important for fine-grained sentiment analysis. Thus, the Semeval-2018 task 1 is aimed to analyze the affective intensity of tweets. In this paper, we introduce the system participating in this task. We apply an attention-based CNN-LSTM model to predict the intensity scores of emotions and sentiment. We also use additional features to improve the performance of our system. Our system ranked 12/48 and 15/38 in the EI-reg and V-reg subtasks respectively. It indicates that our system can be further extended.

Acknowledgments

The authors thank the reviewers for their insightful comments and constructive suggestions on improving this work. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0800402 and in part by the National Natural Science Foundation of China under Grant U1705261, Grant U1536207, Grant U1536201 and U1636113.

References

Francesco Barbieri, German Kruszewski, Francesco Ronzano, and Horacio Saggion. 2016. How cosmopolitan are emojis?: Exploring emojis usage and meaning over different languages with distributional semantics. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 531–535. ACM.

Felipe Bravo-Marquez, Marcelo Mendoza, and Barbara Poblete. 2014. Meta-level sentiment models for big social data analysis. *Knowledge-Based Systems*, 69:86–99.

Tweets with visual attention weights
someone cheer me up
Happy birthday to me ❤ #blessed
What are some good #funny #entertaining #interesting accounts I should follow ? My twitter is dry

Table 6: Visualization of the attention weights of tweets. Red denotes high attention and green denotes low attention.

Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. A feature-enriched neural model for joint chinese word segmentation and part-of-speech tagging. *arXiv preprint arXiv:1611.05384*.

Nadia FF Da Silva, Eduardo R Hruschka, and Estevam R Hruschka Jr. 2014. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, 66:170–179.

Luca Dini and André Bittar. 2016. Emotion analysis on twitter: The hidden challenge. In *LREC*.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153.

Pranav Goel, Devang Kulshreshtha, Prayas Jain, and Kaushal Kumar Shukla. 2017. Prayas at emoint 2017: An ensemble of deep neural architectures for emotion intensity prediction in tweets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 58–65.

Vineet John and Olga Vechtomova. 2017. Uwat-emote at emoint-2017: Emotion intensity detection using affect clues, sentiment polarity and word embeddings. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 249–254.

Aamera ZH Khan, Mohammad Atique, and VM Thakare. 2015. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)*, page 89.

Svetlana Kiritchenko, Saif Mohammad, and Mohammad Salameh. 2016. Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 42–51.

Maximilian Köper, Evgeny Kim, and Roman Klinger. 2017. Ims at emoint-2017: emotion intensity prediction with affective norms, automatically extended resources and deep learning. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–57.

Sreekanth Madisetty and Maunendra Sankar Desarkar. 2017. Nsemo at emoint-2017: an ensemble to predict emotion intensity in tweets. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 219–224.

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

Saif M Mohammad. 2016. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In *Emotion measurement*, pages 201–237. Elsevier.

Saif M Mohammad and Felipe Bravo-Marquez. 2017. Wassa-2017 shared task on emotion intensity. *arXiv preprint arXiv:1708.03700*.

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 Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Finn Årup Nielsen. 2011. A new anew: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Kahlil Philander, YunYing Zhong, et al. 2016. Twitter sentiment analysis: capturing sentiment from integrated resort tweets. *International Journal of Hospitality Management*, 55:16–24.

Cicero dos Santos and Maira Gatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78.

Aliaksei Severyn and Alessandro Moschitti. 2015. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM.

Mike Thelwall, Kevan Buckley, and Georgios Paltoglou. 2012. Sentiment strength detection for the social web. *Journal of the Association for Information Science and Technology*, 63(1):163–173.

Theresa Ann Wilson. 2008. *Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states*. University of Pittsburgh.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *arXiv preprint arXiv:1512.05193*.

Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.