IIITBH at WNUT-2020 Task 2: Exploiting the best of both worlds

Saichethan Miriyala Reddy[*]
Dept. of Computer Science
IIIT Bhagalpur
Bihar, India
miriyala.cse.1725@iiitbh.ac.in

Pradeep Kumar Biswal
Dept. of Computer Science
IIIT Bhagalpur
Bihar, India
pkbiswal.cse@iiitbh.ac.in

Abstract

In this paper, we present IIITBH team's effort to solve the second shared task of the 6th Workshop on Noisy User-generated Text (W-NUT) i.e Identification of informative COVID-19 English Tweets. The central theme of the task is to develop a system that automatically identify whether an English Tweet related to the novel coronavirus (COVID-19) is informative or not. Our approach is based on exploiting semantic information from both max pooling and average pooling, to this end we propose two models.

1 Introduction

COVID-19 pandemic started in Wuhan, China in December 2019, caused by the infection of individuals by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) this dangerous virus is spreading around the world since then. The COVID-19 pandemic continues to have a devastating effect on the health and well-being of the global population. It is creating fear and panic for people all around the world, while the vaccine can hopefully brings the situation under control soon. To track the development of the outbreak and to provide users with the information related to the virus, e.g. any new cases in the user's regions. Need for building real-time monitoring system which uses social network data like Twitter is high. However, manual approaches to identify the informative Tweets require significant human efforts and thus are costly. To help handle this problem, WNUT shared task 2 (Nguyen et al., 2020) aim participants to build systems to automatically identify whether a COVID-19 English Tweet is informative or not. Such informative Tweets provide information about recovered, suspected, confirmed and death cases as well as location or travel history of the cases.

Pooling-based recurrent neural architectures consistently outperform their counterparts without pooling (Maini et al., 2020). However, the reasons for their enhanced performance are largely unexamined. In this work, we examine how two most commonly used pooling techniques (mean-pooling or average pooling, and max-pooling) perform for solving WNUT-2020 shared task 2[1] and develop two novel systems exploiting semantic features of both techniques.

2 Data

Dataset consists a total of $10,000$ tweets split into training, validation, test set in 70/10/20 ratio respectively. Detailed breakdown of data is shown in Table 1. Maximum and minimum length of tweets in test data is 64 and 8 respectively. Distribution of tweet length in test dataset is illustrated in Figure 1

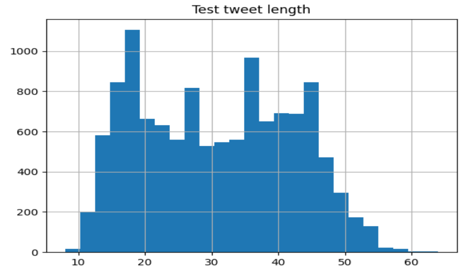

Figure 1: Frequency vs length of tweet

	Informative	Uninformative	Total
Training	3303	3697	7000
Validation	472	528	1000
Test	944	1056	2000
Total	4719	5281	10000

Table 1: Dataset Statistics

[*] Major Contribution and corresponding author

[1] http://noisy-text.github.io/2020/

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 342–346
Online, Nov 19, 2020. ©2020 Association for Computational Linguistics

3 Proposed Methodology

In our proposed architecture, we aim to leverage the semantic information from both pooling layers for identifying whether given tweet is informative or not. In this section, we describe our method (base model illustrated in Figure 2) and elaborate on each part with details.

3.1 Bidirectional LSTM

Recurrent neural network (RNN) is a form of neural network which maintains a memory based on history information. RNNs are good for sequential prediction, but the problem of exploding or vanishing gradients makes learning long distance dependencies very difficult for them (Hochreiter, 1998). The LSTM architecture is proposed to address this problem (Hochreiter and Schmidhuber, 1997). Bidirectional LSTM uses the features coming from both the previous hidden states as well as the future hidden states. This structure allows the networks to have both forward and backward information about the sequence at every time step. It helps the language model in understanding the context better (Schuster and Paliwal, 1997).

Formally, at time t, the memory, c_t, and the hidden state, h_t, are updated with the following equations.

$$i_t = \sigma(W_{xi}h_{t-1} + W_{ci}c_{t-1}) \quad (1)$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot tanh(W_{xc}X_{w,t}W_{hc}h_{t-1}) \quad (2)$$

$$o_t = \sigma(W_{xo}x_{w,t} + W_{ho}h_{t-1} + W_{co}c_t) \quad (3)$$

$$h_t = o_t \odot tanh(c_t) \quad (4)$$

where x is the input at time step t. Bidirectional LSTM contains two separate LSTMs to capture both past and future inputs. One of the LSTM networks encodes the sentence from left to right and the other one from right to left.

$$\overrightarrow{h_t} = Forward(h_t) \quad (5)$$

$$\overleftarrow{h_t} = Backward(h_t) \quad (6)$$

$$h_T = \overrightarrow{h_t} \bigoplus \overleftarrow{h_t} \quad (7)$$

Thus, for each time step t, we obtain two representations, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, finally these two representations are concatenated to form the final output, h_T.

For the purpose of simplifying the information in the output from the Bi LSTM layer (passed through the activation function), pooling layers are used. Pooling layer is a down sampling method, which reduces the number of parameters of the feature map,retaining the important information. Different pooling types like average, max, sum, etc., present. However common pooling types are Max pooling and Average Pooling.

$$S = x_1, x_2, ..., x_n \quad (8)$$

Let S be an input tweet, where x_t is a representation of the input word at position t. A recurrent neural network such as a Bi-LSTM produces a hidden state h_T (equation 7) .

3.2 Average Pooling

Average pooling weighs down the activation by combining the nonmaximal activations (Passricha and Aggarwal, 2019)

$$y^i_{ap} = avg_{i\epsilon(1,n-w)}(h^{i:i+w}) \quad (9)$$

$$y_{ap} = [y^1_{ap}, y^2_{ap}, ..., y^{n-w+1}_{ap}] \quad (10)$$

where w is width of pooling window

$$\xi_{ap} = average(y_{ap}) \quad (11)$$

The use of a global average pooling(ξ_{ap}) layer as a last layer was proposed by (Lin et al., 2013), and got its breakthrough by the well known image recognization system, the residual network (ResNet) (He et al., 2015).

3.3 Max Pooling

Max pooling extracts only the maximum activations (Passricha and Aggarwal, 2019) independent of distribution. One dimensional max pooling can be expressed as follow:

$$y^i_{mp} = max_{i\epsilon(1,n-w)}(h^{i:i+w}) \quad (12)$$

$$y_{mp} = [y^1_{mp}, y^2_{mp}, ..., y^{n-w+1}_{mp}] \quad (13)$$

where w is width of pooling window

$$\xi_{mp} = max(y_{mp}) \quad (14)$$

Global max pooling(ξ_{mp}) was proposed for weakly-supervised learning (Oquab et al., 2014) and is also used in the PHOCNet for the task of word spotting. (Sudholt and Fink, 2016).

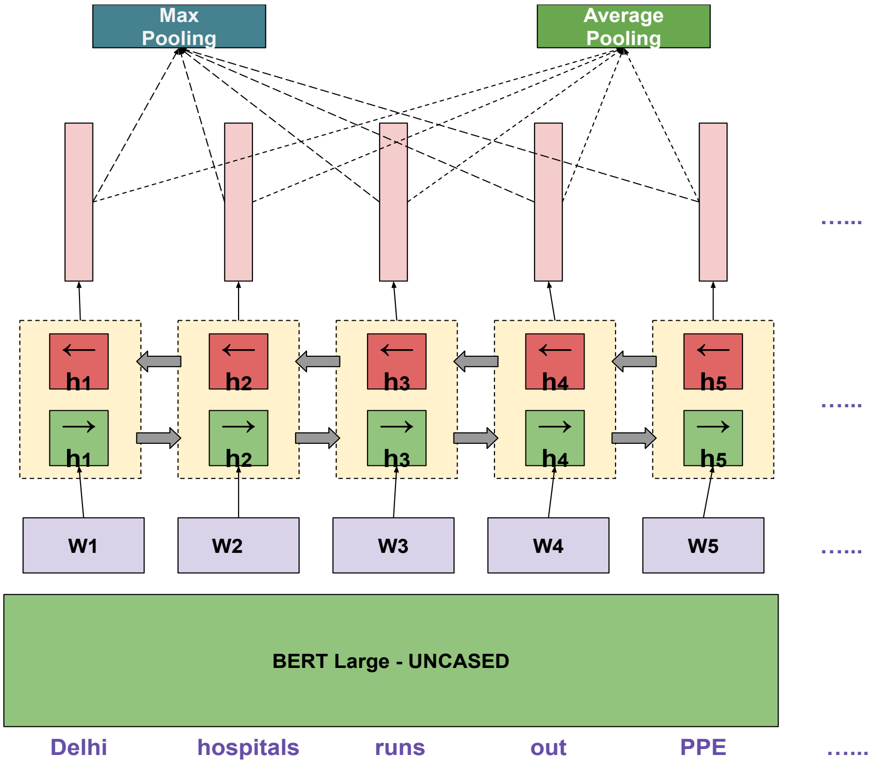

Figure 2: Base Model

From (Section 3.2 and 3.3) we know that max pooling identify only maximum activations irrespective of distribution and frequency, wheras average pooling focus on distribution, and frequency irrespective of maximum values. To leverage this both types of information we propose the following two models (Section 3.4 and 3.5)

3.4 Model I

In Model I, we simply concatenate both global max pooling and global average pooling layers (equation 15). Though this can be considered as a naive model but previous works (Nguyen et al., 2018) (Sun et al., 2018) (Tu et al., 2017) suggests that feature concatenation improves performance of systems, our results supported this intuition.

$$h^* = \xi_{mp} \bigoplus \xi_{ap} \tag{15}$$

where ξ_{mp}, ξ_{ap} are global max pooling and global average pooling from eq 14 and eq 11 respectively.

$$\hat{y} = \sigma(h^*) \tag{16}$$

where, $\sigma(z) = \frac{1}{1+e^{-z}}$.

3.5 Model II

.

In Model II, we intend to use the information such as distribution and frequency from average pooling to understand the context better. While the max-pooling layer attempts to find the most important latent semantic factors in the tweet (Lai et al., 2015). First, we compute the dot product of global average pooling and global max pooling (equation 17), and later multiply with global average pooling (equation 18)

$$o = \xi_{ap} \odot \xi_{mp} \tag{17}$$

$$h^{\dagger} = o \otimes \xi_{ap} \tag{18}$$

where ξ_{mp}, ξ_{ap} are global max pooling and global average pooling from eq 14 and eq 11 respectively.

$$\hat{y} = \sigma(h^{\dagger}) \tag{19}$$

where, $\sigma(z) = \frac{1}{1+e^{-z}}$.

Note: For both models, we used binary cross entropy as our loss function. We submitted results of both systems (Model I & Model II).

4 Experimental Setup

Our model is implemented in Tensorflow[2] and Keras[3]. We use a batch size of B = 500, we train our neural network for 25 epochs with the Adam optimizer. A dropout and recurrent dropout of 0.25 is used. Complete code is made available on Github[4]. During the pre processing stage of data we removed all unwanted symbols and user mentions. Large-uncased BERT model is employed for obtaining tweet embeddings. We also analysed how accuracy and loss of max pooling and average pooling changes with number of epochs in different contextual embeddings (Devlin et al., 2019) (Peters et al., 2018) (Yang et al., 2019) complete code and plots are uploaded in our repository.

5 Results

In order to illustrate the efficacy of our proposed methods, we compare the results with simple average pooling and max pooling on validation set in Table 2. Results in Table 2 are average of 5 runs of each model. From this Table we can infer that our proposed models perform better than existing approaches. In Figure 3 and 4 we illustrated how loss vary with number of epochs on validation data.

Model	F1	Precision	Recall	Accuracy
Avg Pool	84.08	84.28	83.86	85.06
Max Pool	84.18	84.50	84.01	85.13
Model I	84.58	84.73	84.30	85.41
Model II	**84.79**[†]	85.05	84.69	86.04

Table 2: Results on Validation data

From Table 2 we can infer our proposed models (section 3.4 and section 3.5) works than simple average or max pooling. Results of our proposed models on test data are showed in Table 3, we achieved an F1 score of 0.7979 using Model II and 0.7932 using Model I.

Model	Test			
	F1 score	Precision	Recall	Accuracy
Model II	0.7979	0.7991	0.7966	0.8095
Model I	0.7932	0.7983	0.7881	0.8060

Table 3: Results on test data

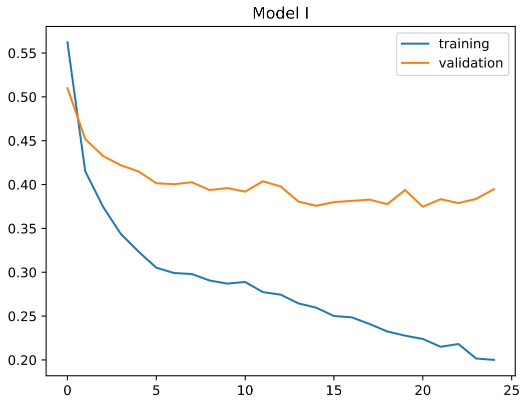

Figure 3: Model I loss on validation set

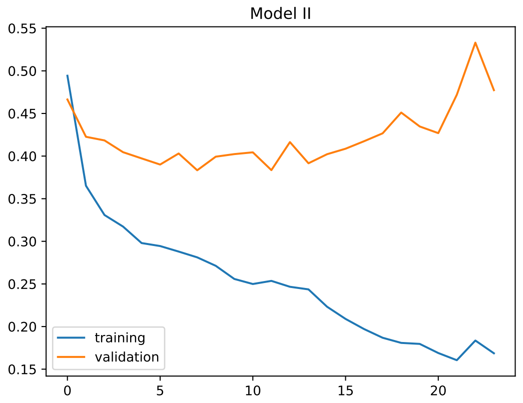

Figure 4: Model II loss on validation set

6 Conclusion

In this paper we presented our system to WNUT 2020 shared task on ”Identification of informative COVID-19 English Tweets”. Traditional text classification models mainly focus on three topics: feature engineering, feature selection and using different types of machine learning algorithms. Our main goal in this paper is to show how we can leverages on different pooling methods of BiLSTM, without using any human-engineered features and improve efficacy of any system. We believe performance of our system can be further improved by tweaking hyper parameters. In future we would like to explore how our models perform with different attention mechanisms (Vaswani et al., 2017) for different tasks like relation classification (Zhou et al., 2016), image captioning (Xu et al., 2015), and machine translation (Bahdanau et al., 2014).

[2] https://www.tensorflow.org/

[3] https://keras.io/

[4] https://github.com/Saichethan/WNUT-2020

Acknowledgments

We would like to thank anonymous reviewer for their suggestion, which helped us in improving the quality of paper further. We would also like to thank Dr. Aravind Choubey (Director), for encouraging research at Indian Institute of Information Technology, Bhagalpur[5].

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 6(2):107–116.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Siwei Lai, L. Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*.

Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network.

Pratyush Maini, Keshav Kolluru, Danish Pruthi, and Mausam. 2020. Why and when should you pool? analyzing pooling in recurrent architectures.

Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*.

L. D. Nguyen, D. Lin, Z. Lin, and J. Cao. 2018. Deep cnns for microscopic image classification by exploiting transfer learning and feature concatenation. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724.

Vishal Passricha and Rajesh Kumar Aggarwal. 2019. End-to-end acoustic modeling using convolutional neural networks. In *Intelligent Speech Signal Processing*, pages 5–37. Elsevier.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Sebastian Sudholt and Gernot A Fink. 2016. Phocnet: A deep convolutional neural network for word spotting in handwritten documents. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 277–282. IEEE.

Xudong Sun, Pengcheng Wu, and Steven CH Hoi. 2018. Face detection using deep learning: An improved faster rcnn approach. *Neurocomputing*, 299:42–50.

Yan-Hui Tu, Jun Du, Qing Wang, Xiao Bao, Li-Rong Dai, and Chin-Hui Lee. 2017. An information fusion framework with multi-channel feature concatenation and multi-perspective system combination for the deep-learning-based robust recognition of microphone array speech. *Computer Speech & Language*, 46:517–534.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention.

Z. Yang, Zihang Dai, Yiming Yang, J. Carbonell, R. Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

[5] https://www.iiitbh.ac.in/