LING 530F Assignment 2
WASSA 2018 Implicit Emotion Shared Task

Meng Li
meng.li@alumni.ubc.ca

Yihao Zhang
yihao1@ece.ubc.ca

Abstract

Deep learning has been a popular topic in the research field of machine learning and artificial intelligence (LeCun et al., 2015; Goodfellow et al., 2016). Researchers have been developing many models based on deep learning disciplines to analyze large amounts of text data (Cho et al., 2014; Graves and Schmidhuber, 2005). Machine learning approach (MLA), specifically deep learning technique are one of the most common approaches of Emotion Detection and Recognition. Implicit Emotion Shared Task (IEST) has been proposed to develop models which can classify a text into one of the following emotions: *anger*, *fear*, *sadness*, *joy*, *surprise*, *disgust* without having access to an explicit mention of an emotion word given the sample training data from twitter. In this paper, such system with long-short-term-memory (LSTM) newtork (Gers et al., 1999) which is a special kind of Recurrent Neural Network (RNN) is built to complete IEST. The purpose is to build such model detect implicit emotion from given tweets and achieve high F_1 scores.

1 Introduction

Emotion has the effect on people. Emotion recognition is the process of identifying human emotion. Facial expression and verbal expressions are two main sources of emotion recognition(Wikipedia contributors, 2004). Emotion recognition has been already applied to various fields of applications such as healthcare (Petrantonakis and Hadjileontiadis, 2010), mobile (Chen et al., 2015), social media (Ekman, 1999) and market research. The growing trend of using machine learning techniques in NLP is observed in recent years. Huge amount of work was done on emotion recognition and detection from text. (Yam, 2015) proposed a trained model for emotion detection and recognition from text. The model is a multiple layer neural network with 3 hidden layers of 125, 25 and 5 neurons respectively. 60% of 784,349 samples were used for training, 20% for testing and 20% for validation. The unweighted accuracy of 64.47% and weighted accuracy of 60.60% were obtained. (IBM, 2016) proposed API called IBM Watson Tone Analyzer for detecting tones from written text. Emotion, language and social tones can be categorised by IBM Watson Tone Analyzer. Qemotion provides the analysis on written text and detect the main emotion. It is claimed by Qemotion that the accuracy of more than 85% is obtained by the semantic algorithm using statistic models. It is possible to transfer the knowledge of emotions to machines. Because emotions depend on contexts within the sentence, the specific word of emotions such as ”anger” does not need to appear in order to tell the emotion of the sentences. Other phrases such as ”shut up” can be the alternatives of ”anger”(Yam, 2015). Another challenge for transferring such knowledge of emotions to machines is large amount of labelled data. It is also applied to general machine learning tasks. We thank IEST organized by WASSA 2018 at EMNLP 2018 that they provides the large training data with/without labels for learning purpose and test data for evaluation. The details and the format of training data will be discussed in the following Data section. The results are shown in Results section and the comparison between performance of LSTM and bidirectional LSTM (biLSTM) model is also included. Possible works are discussed in order to improve the accuracy using different pre-processing methods and models in

Future Work section.

2 Related Work

UBC-NLP has posted the work for same task of IEST 2018 and reached the impressive 70.7% F_1 score(Alhuzali et al., 2018). Similar ways for manipulating the results are used in this paper.

3 Data

Both training data for learning and test data for evaluation are provided by WASSA 2018 IEST.
The training data which is a csv file has 2 columns and 153383 rows. The first column has the emotion which the sentence of corresponding row represents. The second column has the original sentence with emotion word replaced by "[TRIGGERWORD]".
The test data has two columns and 9501 rows. Thus, it has 9501 tweets with corresponding correct labels of emotions. The first column has the fake emotion tag with "joy" and it is discarded when it is processed. The second column is the test sentences with emotion word replaced by "[TRIGGERWORD]". Another test data consists of 9501 corresponding emotion labels for evaluation. The overall F_1 score is computed using test data as a measurement of test's accuracy.

4 Methods

4.1 Pre-processing

We use a very simple schema to pre-process the dataset. For each tweet, all "[TRIGGERWORD]" and "[USERNAME]" words are replaced with empty space. The url links are also removed. All texts are lowercased for simplicity. However, due to the complex task of processing emoji within limited time, we just leave them without any processing, which definitely decrease the accuracy of our model. For the vocabulary, we retain the words which appear at least 3 times, with a total of 24358 words.

4.2 Methods

We develop a neural language model (language decoder) using the Long short term memory (LSTM) networks, based on PyTorch seq2seq tutorial (Robertson, 2017). LSTM networks are capable of classifying, processing and making predictions especially based on time series data, since language sentences can seen as some dependent event happened within some time series. Traditional RNNs have the problem of exploding or vanishing gradients, which can be solved by LSTMs. LSTMs are a special kind of recurrent neural network (RNN) which is capable of learning long-term sequential data dependencies. The use of training gates within a LSTM cell allows the network to learn long-term structure without suffering from the problem of gradient exploding or gradient vanishing. There have been many studies on applying LSTM in natural language processing such as machine translation(Alexey Dosovitskiy and Brox, 2014), English text generation (Baskur) and speech-to-text generation. Relative insensitivity to gap length is another advantage of LSTM over RNNs, and therefore LSTMs are widely used in natural language processing (Wikipedia contributors, 2018).

Hyper-Parameter	Value
word embedding dimension	300
hidden layers	1
hidden layer dimension	300
batch size	1
epochs	2

Table 1: Network architecture and hyper-parameters for our LSTM model.

4.3 Training details

We train our LSTM model for different numbers of epochs. The results are shown in results section. The Adam optimizer and CrossEntropyLoss are used for training phase. We use the teacher forcing to ensure the real target outputs as each next input, instead of using the decoders guess as the next input. Using teacher forcing will cause the model to converge faster. The network architecture and hyper-parameters are shown in Table 1.

5 Results

The result is obtained with 9501 training sets and golden value of corresponding 9501 tags. From the statistical results shown as Figure 1, the number of true positive, false positive, false negative, precision, recall and F_1 scores are computed. The model is trained for 1 epoch, 2 epochs and 10 epochs for comparison. The final F_1 score is 47.479% which is from 10 epochs training for baseline evaluation.

```
Labels: sad;surprise;anger;disgust;joy;fear
Label    TP      FP      FN      P       R       F
sad      397     280     1063    0.586   0.272   0.372
surpris  697     799     903     0.466   0.436   0.45
anger    691     1077    909     0.391   0.432   0.41
disgust  908     1198    689     0.431   0.569   0.49
joy      983     622     753     0.612   0.566   0.588
fear     891     1048    707     0.46    0.558   0.504
MicAvg   4567    5024    5024    0.476   0.476   0.476
MacAvg                           0.491   0.472   0.469
Official result: 0.4691360641845818
```

Figure 1: Evaluation result for 1 epoch training

	anger	disgust	fear	joy	sad	surprise
anger	691	254	231	155	59	210
disgust	205	908	147	78	76	183
fear	236	176	891	91	39	165
joy	174	147	258	983	54	120
sad	248	325	184	185	397	121
surprise	214	296	228	113	52	697

Figure 2: Correctness count for 1 epoch training

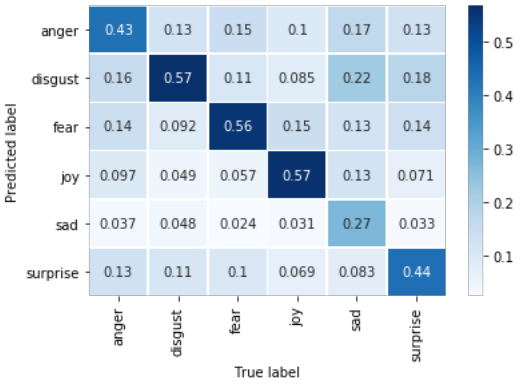

Figure 3: Result as heatmap for 1 epoch training

Figure 2 shows the counting for matched and unmatched labels with 1 epoch training. For better observations based on the test results, the percentage of correctness is computed instead. (Alhuzali et al., 2018) uses heatmap for better visualization of result. The same technique is adopted in this paper. Figure 3 uses heat map is used to visualize percentage of correctness. The rows are golden labels and columns are labelled categories. The diagonal entries are the correct labels. From Figure 3, ”sad” is the most confusing category for our LSTM model. Sentences with label ”sad” are mostly mislabeled as ”disguest”. ”disguest” and ”joy” have the highest correctness of 57%. Since 1 epoch training will possibly lead to underfitting results, our model is training in 2 epochs and 10 epochs to see the improvement with incresing number of epochs. The results are shown correspondingly.

```
Labels: sad;surprise;anger;disgust;joy;fear
Label    TP      FP      FN      P       R       F
sad      453     336     1007    0.574   0.31    0.403
surpris  800     1105    800     0.42    0.5     0.456
anger    657     996     943     0.397   0.411   0.404
disgust  879     1103    718     0.443   0.55    0.491
joy      920     508     816     0.644   0.53    0.582
fear     861     973     737     0.469   0.539   0.502
MicAvg   4570    5021    5021    0.476   0.476   0.476
MacAvg                           0.491   0.473   0.473
Official result: 0.47296008802830075
```

Figure 4: Evaluation result for 2 epochs training

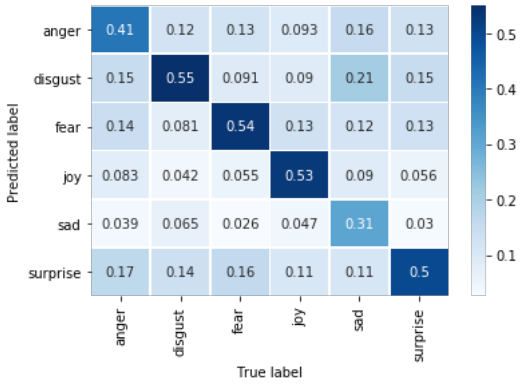

Figure 5: Result as heatmap for 2 epochs training

Figure 4 and Figure 5 show the training result of 2 epochs. Training with 2 epochs gives slightly better result in F_1 score of 47.296%. The correctness of labelling ”sad” is also improved.

```
Labels: sad;surprise;anger;disgust;joy;fear
Label    TP      FP      FN      P       R       F
sad      467     316     993     0.596   0.32    0.416
surpris  784     1086    816     0.419   0.49    0.452
anger    622     850     978     0.423   0.389   0.405
disgust  822     910     775     0.475   0.515   0.494
joy      990     628     746     0.612   0.57    0.59
fear     912     1204    686     0.431   0.571   0.491
MicAvg   4597    4994    4994    0.479   0.479   0.479
MacAvg                           0.493   0.476   0.475
Official result: 0.4747540503681784
```

Figure 6: Evaluation result for 10 epochs training

Figure 6 and Figure 7 give result of training with 10 epochs. Training with 10 epochs improves the accuracy to 47.48% and it is the best result we get with current work. At this point, further increasing the number of epochs seems not efficient enough. Other ways which may increase the accuracy are discussed in the following section.

We also change the model into biLSTM to compare with original LSTM model. The difference

Figure 7: Result as heatmap for 10 epochs training

between biLSTM and LSTM is that the learning algorithm will be fed with the original data once from beginning to the end and once from end to beginning. Figure 8 and Figure 9 show the result for biLSTM model. However, LSTM gives slightly better result in this case.

```
Labels: sad;surprise;anger;disgust;joy;fear
Label    TP      FP      FN      P       R       F
sad      286     223     1174    0.562   0.196   0.291
surpris  929     1913    671     0.327   0.581   0.418
anger    499     810     1101    0.381   0.312   0.343
disgust  887     1543    710     0.365   0.555   0.441
joy      761     351     975     0.684   0.438   0.534
fear     686     703     912     0.494   0.429   0.459
MicAvg   4048    5543    5543    0.422   0.422   0.422
MacAvg                          0.469   0.419   0.414
Official result: 0.4143527272333148
```

Figure 8: Evaluation result for BiLSTM training

Figure 9: Result as heatmap for BiLSTM training

6 Future work

It is a reasonable assumption that the model performance can be improved with future work. The model is using gradient descent which is an iterative learning process based on a limited dataset to optimize learning. A training process which uses not enough epochs will lead to underfitting results. From the result of current work, there is a trend of growing accuracy with more epochs. To future improve the accuracy, more epochs can be adopted. Normally MSE will decrease with numbers of epochs at the beginning. Increasing number of epochs would possibly improve the test result. However, since increasing number of epochs only improves the accuracy slightly and one epoch takes around 10 hours to finish with selected hyperparameters of hidden layer dimension and word embedding dimension, other ways need to be considered preferentially.

Another way to improve the model is using bidirectional LSTM model. Compared to the original LSTM model, bidirectional LSTM allows the backward direction to preserve the information from future.

Better pre-processing of raw data is another possible way to improve the model. For instance, Our current work only includes removing "[TRIGGERWORD]", "[USERNAME]" and url. Emoji is another type which contains huge amount of information regarding emotion. Translating emoji into corresponding words will probably improve the model.

Adding more hidden layers or increasing number of hidden neurons into LSTM model is also possible to improve the test result. Our model contains one hidden layer of 300 hidden neurons. With more hidden layers or neurons added, it is reasonable to assume that it will help reduce underfitting and improve the model accuracy.

7 Conclusion

The present paper aims to detect implicit emotion from twitter data. We described an LSTM based neural network system detecting implicit emotion from twitter data sets. Both LSTM and BiLSTM models are constructed using PyTorch and evaluated with baseline evaluation proposed by WASSA 2018. However, the performance of LSTM and biLSTM model are similar for such task. The highest F_1 score among different models and number of epochs is 47.48% with LSTM and 10 epochs. The learning process is observed and accruacy increases with epoch numbers. Different ways to improve the model performance are also discussed in the paper.

Acknowledgments

References

Jost Tobias Springenberg Martin Riedmiller Alexey Dosovitskiy and Thomas Brox. 2014. Discriminative unsupervised feature learning with convolutional neural networks https://papers.nips.cc/paper/5548-discriminative-unsupervised-feature-learning-with-convolutional-neural-networks.pdf.

Hassan Alhuzali, Mohamed Elaraby, and Muhammad Abdul-Mageed. 2018. Ubc-nlp at iest 2018: Learning implicit emotion with an ensemble of language models .

Moez BaccoucheFranck MamaletChristian WolfChristophe GarciaAtilla Baskur. ???? Sequential deep learning for human action recognition .

Min Chen, Yin Zhang, Yong Li, Shiwen Mao, and Victor CM Leung. 2015. Emc: Emotion-aware mobile cloud computing in 5g. *IEEE Network* 29(2):32–38.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion* pages 45–60.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with lstm .

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5-6):602–610.

IBM. 2016. "ibm watson tone analyzer". https://www.ibm.com/watson/services/tone-analyzer/.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521(7553):436.

Panagiotis C Petrantonakis and Leontios J Hadjileontiadis. 2010. Emotion recognition from eeg using higher order crossings. *IEEE Transactions on Information Technology in Biomedicine* 14(2):186–197.

Sean Robertson. 2017. Pytorch seq2seq tutorial. https://pytorch.org/tutorials/intermediate/seq2seq_translation_tutorial.html. Accessed: 2010-09-30.

Wikipedia contributors. 2004. Emotion recognition — Wikipedia, the free encyclopedia. [Online; accessed 22-July-2004]. https://en.wikipedia.org/wiki/Emotion$_r ecognition$.

Wikipedia contributors. 2018. Long short-term memory — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Long_short-term_memory&oldid=862158109. [Online; accessed 21-October-2018].

CY Yam. 2015. Emotion detection and recognition from text using deep learning https://www.microsoft.com/developerblog/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning/.