Sentiment Analysis with SemEval Dataset

An exploration of factors affecting training effects

Xianglong Hu
xh1012
New York University

Bingqing Deng
bd1394
New York University

ABSTRACT

This paper describes the fifth year of the *Sentiment Analysis in Twitter Task 4 Subtask A*. Task 4 includes the overall sentiment of the tweet, sentiment toward a topic with classification on a two-point and five-point ordinal scale. We explore subtask A, analyzing the overall sentiment on a three-point scale. Following a neural network methodology, the problem is modeled on the state-of-art bidirectional LSTM(long short-term Memory) architecture. This paper emphasizes the effort to explore different factors that affect the accuracy of the neural network model instead of achieving cutting edge results. The code is written in Pytorch, which is the most popular and pythonic deep learning framework these days. Relevant techniques and research results are also reviewed. In short, we made two contributions in this paper. One is to experiment with neural network architectures, which is beyond the scope of this course, and the other is to detail the relevant background and machine learning framework for pedagogical purposes. The LSTM finally achieved relatively good results compared to all the teams in SemEval 2017 within a short period of time, with the final accuracy around 65.18%, which is close to the top-one acccuracy 64.6%. However, the model's performance is not robust to parameter perturbations.

1 INTRODUCTION

Neural Network machinery has achieved extraordinary performance in various fields including natural language processing. Two major kinds of neural network architecture that can be combined in various ways dominates in this field, feed-forward networks and recurrent networks. Among these, recurrent neural networks are specialized in sequential data. This perfectly solves the inborn problem of natural language dataset, that is, they are sequential with indefinite length. There are many variations of RNNs to solve practical problems such as vanishing gradients [Goldberg and Hirst 2017]. One prototype of these is called gated architecture. The Long Short-Term Memory architecture is the first to introduce this gated mechanism. Another simpler architecture is Gated Recurrent Unit(GRU), which is less computationally expensive. Since most of the prior work [Rosenthal et al. 2017] [Baziotis et al. 2017] [Cliche 2017] adopted LSTM, we are mainly devoted to this architecture in the paper.

The Sentiment Analysis in Twitter task has been run yearly at SemEval since 2013 [Rosenthal et al. 2017], which is an ongoing series of evaluations of computational semantic analysis systems. This is exactly the reason why we choose this dataset, since we've got a lot of benchmarks to compare with. 37 teams participated subtask A "Message Polarity Classification". Most of them adopt ensemble systems more than a single model. We only adapt a single model here.

Nowadays, there are a lot of machine learning frameworks such as TensorFlow, Caffe, THeano, etc. However, the newly released pytorch is among the most welcomed deep learning frameworks. Thanks to pytorch [Paszke et al. 2017], we are able to implement complicated neural network architecture in a simple, intuitive and pythonic way.

2 ARCHITECTURE AND PREPROCESSING

Following [Baziotis et al. 2017], our network is composed of a single layer bidirectional LSTM followed by a single layer full connected layer, a ReLU and a softmax output layer. This model is actually the simplest LSTM architecture one can come up with. However, we also experiment with other architectures as well. This would be discussed in detail in 4.

[Baziotis et al. 2017] has extra components, a embedding layer and an attention layer, shown in 1. They've achieved 65.1% accuracy, ranked No.5 based solely on accuracy yet No.1 based solely on recall.

We do the data processing as described in [Baziotis et al. 2017]. Namely, all the emojis are replaced and special tokens like user are directly replaced. For simplicity, an example is posted in 2. As to the token embeddings, we used the pretrained model provided by Glove.

3 PERFORMANCE

In 1, it shows the statistics of various dataset we've used. This provides a random guess baseline for evaluating the performance. It is obvious that in each dataset, LSTM gets a much higher chance than random guesses. The LSTM model achieves as high as 59% on incomplete test data of SemEval 2013, probably ranked around the upper half back to 2013 or 2016 since their highest F score is only 0.62. The LSTM model is evaluated on SemEval 2013 and SemEval 2016 test data. For 2013, it is trained on train data. As to 2016, there is a huge gap in the number of training and test data in 1. We don't come up with a good strategy to eliminate the inbalance of test and training data. And after all, no parameter yield good results. So we decide to split 2016 test data into train and development data to indicate the quality of the LSTM Model. Due to the limited computing resources, we are not able to leverage all the data since 2013 to train as other teams do. Since the revealed results on 2013 test is based on F1-score rather than accuracy, it is very hard to directly judge the LSTM's performance on 2013 data. However, these two measures are usually close. Nevertheless, according to [Nakov et al. 2016], 65.61% is beyond the best of 34 teams, which means the results are comparable to the upper quarter back to 2016. As least this means LSTM is good at capturing the feature of this

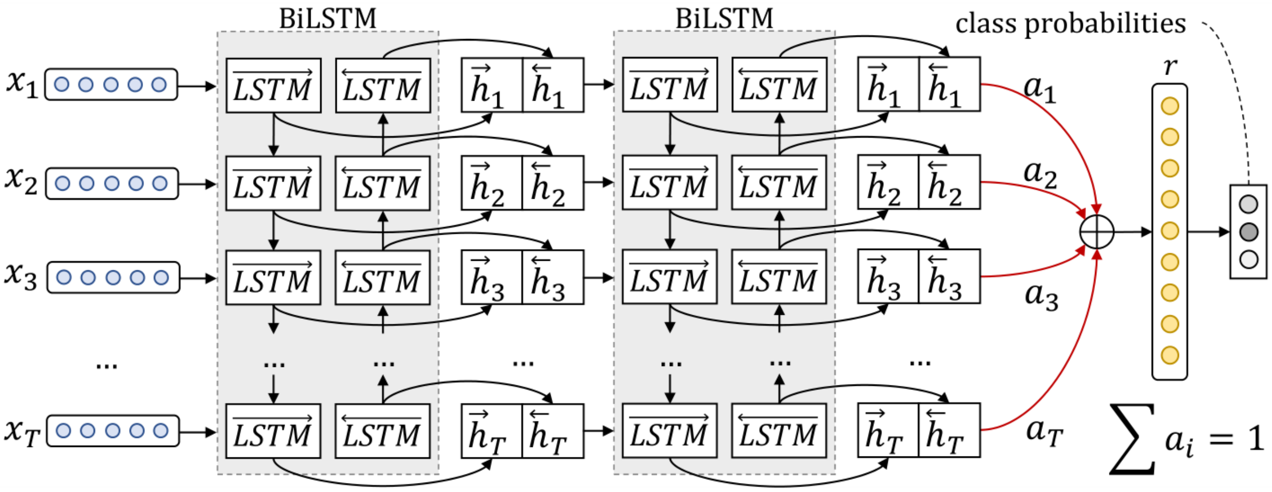

BiLSTM
BiLSTM
class probabilities
x_1
x_2
x_3
x_T
$\overrightarrow{LSTM}$
$\overleftarrow{LSTM}$
$\overrightarrow{h}_1$ $\overleftarrow{h}_1$
$\overrightarrow{h}_2$ $\overleftarrow{h}_2$
$\overrightarrow{h}_3$ $\overleftarrow{h}_3$
$\overrightarrow{h}_T$ $\overleftarrow{h}_T$
a_1
a_2
a_3
a_T
r
$\sum a_i = 1$

Figure 1: Architecture of Team DataStories

original	The *new* season of #TwinPeaks is coming on May 21, 2017. CANT WAIT \o/ !!! #tvseries #davidlynch :D
processed	the new <emphasis> season of <hashtag> twin peaks </hashtag> is coming on <date> . cant <allcaps> wait <allcaps> <happy> ! <repeated> <hashtag> tv series </hashtag> <hashtag> david lynch </hashtag> <laugh>

Figure 2: Example of Sentence Processing

Table 1: Performance on SemEval 2013 and 2016

dataset	negative	neutral	positive	performance	size
SemEval 2013 train	15.05%	47.33%	37.62%	80.14%[a]	9728
SemEval 2013 test	15.77%	42.68%	41.55%	59.06%	3813
SemEval 2016 train	51.57%	14.38%	15.66%	-	6000
SemEval 2016 test	34.21%	50.13%	15.66%	65.18%[b]	20632

Note: This table describes the statistics of data as well as the final results.

[a] This is accuracy during training.

[b] This is development accuracy when using test data as train data.

dataset. Considering our limited training dataset as well as limited computing resources, this is actually a quite good result.

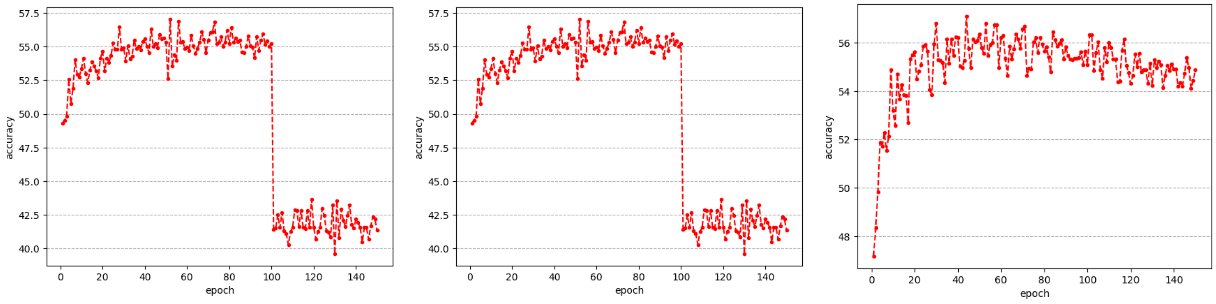
4 EXPERIMENTS

Various factors of experiments have been researched. They can be divided into two kinds of factors, architecture design and prior parameters adjustment. In the following session, the experiments are done with SemEval 2013 data without explicit clarification.

As to the design of the architecture, we have done experiments on single layer bidirectional LSTM, bilayer LSTM as well as the triple layer one. It is rather surprising that the accuracy gets worse as the number of the layers grow, which is shown in 3. This is probably due to the shortage of training data. As seen in 1, there are only less than 10000 pieces of data available. However, the parameters of LSTM grows linearly with number of layers. A single layer model has 29683 weights, while the two has 68723 and the three 107763. The number of parameters are far more than the data we have.

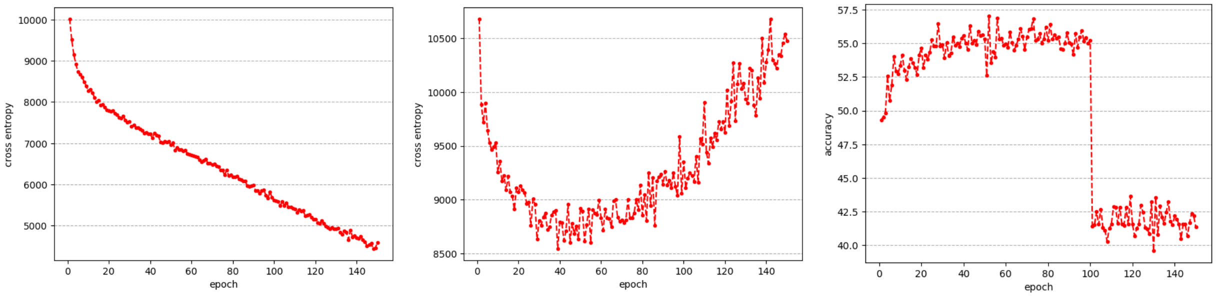
One plausible solution is to use dropout, we have tried to set dropout rate as 0.5. However, it doesn't seem to have much improvement.

We have also tried to use epoch varying learning rate since 4c is observed. However, this doesn't seem to work. The learning rate gets multiplied by 0.1 every 50 epochs. In convex problems, making the learning rate smaller always works but this is not the case in this problem.



(a) single layer **(b) two layers** **(c) three layers**

Figure 3: Effect of layer number



(a) training loss **(b) testing loss** **(c) testing loss**

Figure 4: Overfitting and Accuracy Drop

There is a significant overfitting problem as seen in 4a and 4b. However, this also reflects some intrinsic conflicts between training and testing data.

5 CONCLUSION

We have achieved satisfying results on SemEval 2013 Dataset with LSTM. We’ve also experimented with various architecture and parameter settings to explore their impact on final results. It turns out that when trapped in a local optimum, it is hard to design a mechanism to get rid of it. It turns out that LSTM is not very robust to prior parameters. It usually takes a long time to figure out a suitable parameter. Nonetheless, there is no mechanism for us to find best priors. However, LSTM are successful somehow in achieving much better results than baseline random guesses.

REFERENCES

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 11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, Vancouver, Canada, 747–754.

Mathieu Cliche. 2017. BB_twtr at SemEval-2017 Task 4: Twitter Sentiment Analysis with CNNs and LSTMs. *CoRR* abs/1704.06125 (2017). arXiv:1704.06125 http://arxiv.org/abs/1704.06125

Yoav Goldberg and Graeme Hirst. 2017. *Neural Network Methods in Natural Language Processing*. Morgan & Claypool Publishers.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. 2016. SemEval-2016 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval ’16)*. Association for Computational Linguistics, San Diego, California.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. (2017).

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval ’17)*. Association for Computational Linguistics, Vancouver, Canada.