Sentiment Analysis for Movie Reviews

Mohamed Nada, Nile University, mo.nada@nu.edu.eg

***Abstract* - Presenting approaches that use unlabeled data to improve sequence learning with recurrent networks. The parameters obtained from the unsupervised step can be used as a starting point for the supervised training model. Experiments showed that long short-term memory recurrent networks after being pretrained are more stable and generalize better. With pretraining, we are able to train long short-term memory recurrent networks up to a few hundred timesteps, thereby achieving strong performance in many texts classification tasks, such as IMDb.**

Index Terms – Long Short-Term Memory (LSTM), Internet Movie Database (IMDb).

INTRODUCTION

Sentiment analysis is an NLP technique that identifies the polarity of a given text. The model should classify inputs to positive, negative, and neutral. Sentiment analysis is used in a wide variety of applications, for example: Social Media Mentions, Product Reviews, and Complaint Tickets.

There is ready Pre-trained models for each type of applications.

PROBLEM STATEMENT

Analyze Text Sentiments to:

- Understand how people are talking about movies compared to each other's.
- Optimize recommendation system
- Quickly get insights into what people like and dislike about movies.

MOTIVATION

Optimizing the sentiment analysis model to get higher accuracy.

PROPOSED SOLUTION

Low performance mainly happened because random initialization. To achieve good performance:

- Careful tuning of hyperparameters.
- Pretraining step.

A simple pretraining step (unsupervised) can significantly stabilize the training of LSTMs.

The parameters obtained from the unsupervised step can be used as a starting point for other supervised training models like LSTM recurrent network Model (sentiment analysis classification task).

DATASET

The IMDb Movie Reviews dataset is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb) labeled as positive or negative. The dataset contains an even number of positive and negative reviews. Only highly polarizing reviews are considered. A negative review has a score ≤ 4 out of 10, and a positive review has a score ≥ 7 out of 10. No more than 30 reviews are included per movie. The dataset contains additional unlabeled data.

Observations :

a) Mean review length = around 69.

b) minimum length of reviews is 2.

c)There are quite a few reviews that are extremely long, we can manually investigate them to check whether we need to include or exclude them from our analysis.

EVALUATION METRICS

As a classification problem, Sentiment Analysis uses the evaluation metrics of Precision.

In an imbalanced classification problem with more than two classes, precision is calculated as the sum of true positives across all classes divided by the sum of true positives and false positives across all classes.

Will compare result of randomly initialized LSTM model vs result of LSTM with pretrained initialization step.

EXPERIMENTS

In this first set of experiments, we benchmark our methods on the IMDB movie sentiment dataset. There are 25,000 labeled and 50,000 unlabeled documents in the training set and 25,000 in the test set. We use 15% of the labeled training documents as a validation set. The average length of each document is 241 words, and the maximum length of a document is 2,526 words. The previous baselines are bag-of-words, ConvNets or Paragraph Vectors. Since the documents are long, one might expect that it is difficult for recurrent networks to learn. We however, find that with tuning, it is possible to train LSTM recurrent networks to fit the training set.

For example, if we set the size of hidden state to be 512 units and truncate the backprop to be 400,

an LSTM can do well. With random embedding dimension dropout and random word dropout (not published previously), we are able to reach performance of around 86.5% accuracy in the test set, which is approximately 5% worse than most baselines.

Fundamentally, the main problem with this approach is that it is unstable: if we were to increase the number of hidden units or to increase the number of backprop steps, the training breaks down very quickly: the objective function explodes even with careful tuning of the gradient clipping. This is because LSTMs are sensitive to the hyperparameters for long documents. In contrast, we find that the SA-LSTM works better and is more stable. If we use the sequence autoencoders, changing the size of the hidden state or the number of backprop steps hardly affects the training of LSTMs. This is important because the models become more practical to train.

Using sequence autoencoders, we overcome the optimization instability in LSTMs in such a way that it is fast and easy to achieve perfect classification on the training set. To avoid overfitting, we again, use input dimension dropout, with the dropout rate chosen on a validation set. We find that dropping out 80% of the input embedding dimensions works well for this dataset. The results of our experiments are shown below together with previous baselines. We also add an additional baseline where we initialize a LSTM with word2vec embeddings on the training set.

Performance of models on the IMDB sentiment classification task.

Model with Test error rate:

- LSTM with tuning and dropout 13.50%
- LSTM initialized with word2vec embeddings 10.00%
- LM-LSTM 7.64%
- SA-LSTM 7.24%
- SA-LSTM with linear gain 9.17%
- SA-LSTM with joint training 14.70%
- Full+Unlabeled+BoW 11.11%
- WRRBM + BoW (bnc) 10.77%
- NBSVM-bi (Na¨ıve Bayes SVM with bigrams) 8.78%
- seq2-bown-CNN (ConvNet with dynamic pooling) 7.67%
- Paragraph Vectors 7.42%

The results confirm that SA-LSTM with input embedding dropout can be as good as previous best results on this dataset. In contrast, LSTMs without sequence autoencoders have trouble in optimizing the objective because of long range dependencies in the documents.

Using language modeling (LM-LSTM) as an initialization works well, achieving 8.98%, but less well compared to the SA-LSTM. This is perhaps because language modeling is a short-term objective, so that the hidden state only captures the ability to predict the next few words.

In the above table, we use 1,024 units for memory cells, 512 units for the input embedding layer in the LM-LSTM and SA-LSTM. We also use a hidden layer 512 units with dropout of 50% between the last hidden state and the classifier. We continue to use these settings in the following experiments except with 30 units in the final hidden layer.

Below, we present some examples from the IMDB dataset that are correctly classified by SALSTM but not by a bigram NBSVM model. These examples often have long-term dependencies or have sarcasm that is difficult to find by solely looking at short phrases.

IMDB sentiment classification examples that are correctly classified by SA-LSTM and incorrectly by NBSVM-bi.

Text with Sentiment in **bold**:

- This film is not at all as bad as some people on here are saying. I think it has got a decent horror plot and the acting seem normal to me. People are way over-exagerating

what was wrong with this. It is simply classic horror, the type without a plot that we have to think about forever and forever. We can just sit back, relax, and be scared. **Positive**

- Looking for a REAL super bad movie? If you wanna have great fun, don’t hesitate and check this one! Ferrigno is incredibly bad but is also the best of this mediocrity. **Negative**

- A professional production with quality actors that simply never touched the heart or the funny bone no matter how hard it tried. The quality cast, stark setting and excellent cinemetography made you hope for Fargo or High Plains Drifter but sorry, the soup had no seasoning...or meat for that matter. A 3 (of 10) for effort. **Negative**

- The screen-play is very bad, but there are some action sequences that i really liked. I think the image is good, better than other romanian movies. I liked also how the actors did their jobs. **Negative**

Our first observation is that it is easier to train LSTMs on this dataset than on the IMDB dataset and the gaps between LSTMs, LM-LSTMs and SA-LSTMs are smaller than before. This is because movie reviews in Rotten Tomatoes are sentences whereas reviews in IMDB are paragraphs.

As this dataset is small, our methods tend to severely overfit the training set. Combining SA-LSTMs with 95% input embedding and 50% word dropout improves generalization and allows the model to achieve 20.3% test set error. Further tuning the hyperparameters on the validation set yields 19.3% test error.

To better the performance, we add unlabeled data from the IMDB dataset in the previous experiment and Amazon movie reviews to the autoencoder training stage.3 We also run a control experiment where we use the pretrained word vectors trained by word2vec from Google News.

FUTURE WORK

- Running a hyperparameter search to optimize your configurations.
- Using pretrained word embeddings like Glove word embeddings
- Increasing the model complexity like adding more layers/ using bidirectional LSTMs

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my teacher Dr. Mohamed Naiel as well as TA Ammar Sherif who gave me the golden opportunity to do this wonderful project on the topic (Sentiment Analysis), which also helped me in doing a lot of Research and I came to know about so many new things, Also I am really thankful to my colleague Mazen Al-Asali for his help getting started in code implementation.

REFERENCES

Dai, Andrew M. and Quoc V. Le. “Semi-supervised Sequence Learning.” NIPS (2015).

AUTHOR INFORMATION

Mohamed Nada, Student, Big data diploma, NileUniversity.