CMPT 413

Performing Sentiment Analysis on Tweets and Movie Reviews

Trevor Gale, Lucky Sivagurunathan, Casey Chou

Sentimentalists

Goals

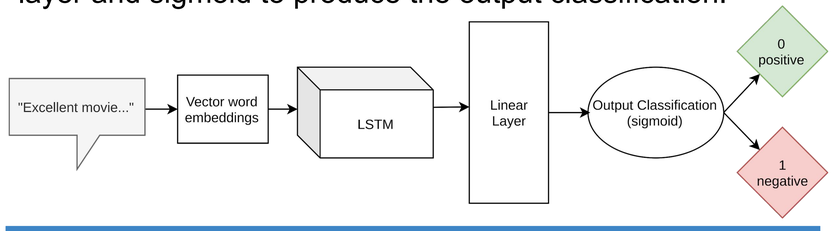
We aim to train a sentiment analysis model on IMDB movie reviews. We will then test the model's effectiveness at sentiment analysis on movie reviews and compare its accuracy with other existing language models in the literature. To test its robustness, the model will also be tested with a Twitter comment dataset available through Kaggle[9].

Previous Work

Previous work on the IMDb dataset for binary sentiment analysis had a state of the art score of 96.21 accuracy[1]. For the domain adaptation task of testing a model trained on IMDb data and applied to Twitter tweets for sentiment analysis using accuracy as the measure we didn't find any previous work for direct comparison. Some domain adaptation work was done in the field using f-score as the metric[5].

Model

We convert movie reviews into a matrix representation by using vector embedding for each word in the review. We then pass this vector into a bidirectional LSTM. The final hidden layers of this LSTM is then passed through a linear layer and sigmoid to produce the output classification.



References

1. Yang, Zhilin, Dai, Zihang, Yang, Yiming, Carbonell, Jaime, Salakhutdinov, Ruslan and Le, Quoc V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. (2019). , cite arxiv:1906.08237 Comment: Pretrained models and code are available at https://github.com/zihangdai/xlnet
2. Andrew L. Maas , Raymond E. Daly , Peter T. Pham , Dan Huang , Andrew Y. Ng , Christopher Potts. Learning word vectors for sentiment analysis, Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, June 19-24, 2011, Portland, Oregon https://pdfs.semanticscholar.org/c521/80a8fe1acc99b4bf3cf3e11d3c8a38e2c7ff.pdf
3. "State-of-the-Art Table for Sentiment Analysis on IMDb." Papers With Code : the latest in machine learning. Accessed December 7, 2019. https://paperswithcode.com/sota/sentiment-analysis-on-imdb.
4. Bentrevett. "Bentrevett/Pytorch-Sentiment-Analysis." GitHub. Accessed December 7, 2019. https://github.com/bentrevett/pytorch-sentiment-analysis/blob/master/2 - Upgraded Sentiment Analysis.ipynb.
5. VMK Peddinti, P Chintalapoodi (2011). "Domain adaptation in sentiment analysis of Twitter," ACM Digital Library. [Online]. Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence,. Available: https://dl.acm.org/citation.cfm?id=2908638. [Accessed: 07-Dec-2019].
6. Pelaez, Alejandro, Talal Ahmed, and Mohsen Ghassemi. "Sentiment Analysis of IMDb Movie Reviews." Accessed December 6, 2019. https://pdfs.semanticscholar.org/c521/80a8fe1acc99b4bf3cf3e11d3c8a38e2c7ff.pdf.
7. Liu, Bing, and Minqing Hu. Opinion Mining, Sentiment Analysis, Opinion Extraction. Accessed December 7, 2019. https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon.
8. Sarkar, Anoop (2019). *Natural Language Processing* Retrieved from: http://anoopsarkar.github.io/nlp-class/assets/slides/nlm.pdf
9. Ripamonti, Paolo. "Twitter Sentiment Analysis." Kaggle. Kaggle, January 2, 2019. https://www.kaggle.com/paoloripamonti/twitter-sentiment-analysis/data.

Implementation and example

Movie Review:

Input movie reviews are converted to a vector representation where each word is represented by an embedding:

```
"A horrible, horrible, horrible film. I saw the original when I was a kid and it gave me nightmares into my teens..."
```

Stop words are removed and stemming is applied:

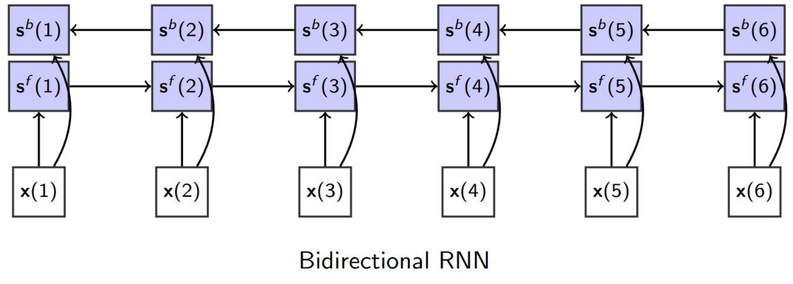
```
"horrible horrible horrible film saw original kid gave nightmares teens..."
```

Vector embedding for "horrible":

```
[3.611810e-02,  3.192400e-03, 7.208750e-02,
-1.080511e-01,  -2.191347e-01,5.814600e-02,
-8.235000e-02,  1.109138e-01, 5.284600e-02,
-4.925090e-02,  5.027700e-02,   2.124790e-02,...]
```

This is done for each word in the review. All reviews are padded or cut so that their lengths are equal.

It is then passed into a bidirectional LSTM [8]:



Bidirectional RNN

The final hidden states are then passed through a Linear layer in order to produce an output classification:

```
0.929043710231781 - negative sentiment
```

Tweet:

The same model can then be run on a negative tweet like:

```
Up bright and early to study after a two hour slumber  but the sun is shining.
```

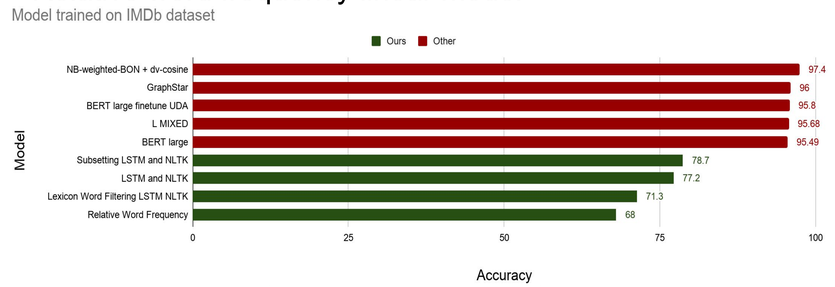
In this case the model scores:

```
0.33244842290878296 - positive sentiment
```

Experimental Evaluation

To evaluate the experiment we used accuracy as the main metric to choose from the following models:

- Subsetting, NLTK, and LSTM (our best model)
- LSTM and NLTK
- Lexicon word filtering, NLTK, and LSTM
- Relative word frequency linear model



	Model trained on IMDB data on IMDB test set	Model trained on IMDB data on twitter test set
Accuracy	78.7%	58.8%

Analysis of Output

Using our best IMDB trained model performed at 78.7%; when applied to twitter data for sentiment analysis the model scored only 58.8%. Tweets have a different level of formality compared to movie reviews, which likely explains the disparity. In particular, tweets are limited to a fixed maximum length of 280 characters. Sentiment in tweets can also be expressed through other means such as emoticons.

We have compared the accuracy of the model with varying training data sizes and iterations and graphed how the model performed when tested on the 4k IMDb dataset.