Machine Learning Final Project

Sentiment Analysis

Group 13

Teammate

M11015802 Eng Tze Qian
M11015080 湯傑堯
M11015018 陳彥家

Introduction

Our goal is to use machine learning methods to find out the sentiment of a sentence is positive or negative. For example :

- This movie is the best .
- This movie is okay.
- This movie is terrible.

According to these three sentences, we can easily know that the first sentence is positive, the second one is neutral, and the last one is negative. As a result, our research will focus on which model will have the best performance in predicting the sentiments.

We have several steps to reach our goal :

- Dataset
- Data Pre-processing
- Building Model
- Training
- Prediction
- Evaluation

Dataset

We use UCI Sentiment labelled sentences dataset(https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences) and ACL IMDb dataset(http://ai.stanford.edu/~amaas/data/sentiment/aclImdb_v1.tar.gz). UCI dataset has three types of data, namely Amazon, IMDb, and Yelp. Each of them has 1000 data. We split them into training data, validation data, and testing data. The ratio is 80%, 10%, 10%. We split the ACL IMD with the same ratio as well.

XLNet

We have five steps in fine-tuning the XLNet model. First is text cleaning, we remove punctuation and numbers from the sentences, and load the pre-train model from google. The next step is tokenizer and encode, which converts the word into number, sentence to sequence. Then we fit our data into the XLNet model. The last step is output and predict the result. The *figure1.1* below shows how we train and fine-tune the XLNet model.

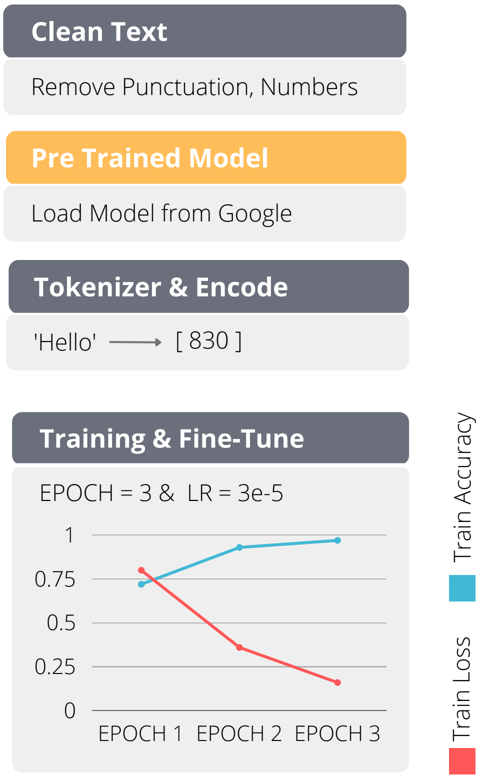


figure1.1

Experiment

- UCI Sentiment Labelled Sentences Dataset.

  For the UCI dataset, we try machine learning methods like SVM and Naive Bayes; Deep learning methods like CNN; Transfer learning methods like ULMFit and XLNet. Figure *1.2* is the structure of our program for machine learning and deep learning.

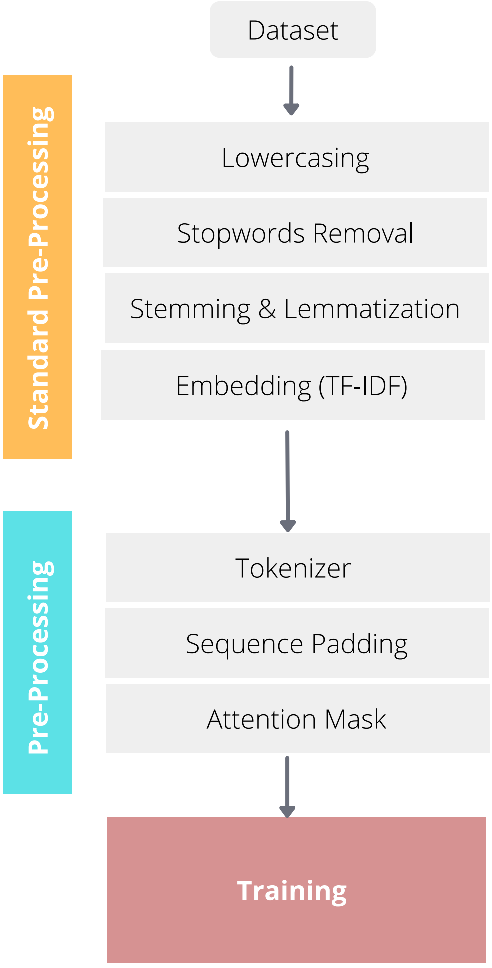


figure1.2

With the structure of program. We change the model to evaluate the result. For the transfer learning model, We follow *figure 1.1* to train our own dataset.

- Result

  Figure 1.3 shows the result of the UCI Sentiment Labelled Sentences Dataset. We used TF-IDF in SVM, Naive Bayes, and CNN. From the figure, we can see that the models do not perform well in separate datasets, since the training data is small, (only 800). However, when combining the three datasets, the performance becomes better, except for CNN, which may be caused by overfitting.

  We can see that ULMFiT performs not well in separate datasets, but have better performance when the dataset is combined. We assume that the model has fewer data to learn from, hence less chance to pick up the general trends in the data that generalized well to new data.

  XLNet performs the best on all three datasets, and the combined one. This is because the XLNet is pre-trained on huge datasets and therefore fine-tuning XLNet will easily get better performance although the training data is small. We also noticed that XLNet performs well on IMDb data, this is probably due to the data there were used to pre-train on XLNet having the same feature as IMDb data.

	Amazon	IMDb	Yelp	Combined
SVM	81.9%	74.5%	78.5%	81.5%
Naive Bayes	78.9%	78.1%	80.0%	82.2%
CNN	71.0%	81.0%	74.0%	75.0%
ULMFiT	77.8%	74.2%	76.8%	84.4%
XLNet	89.5%	95.5%	89.3%	92.7%

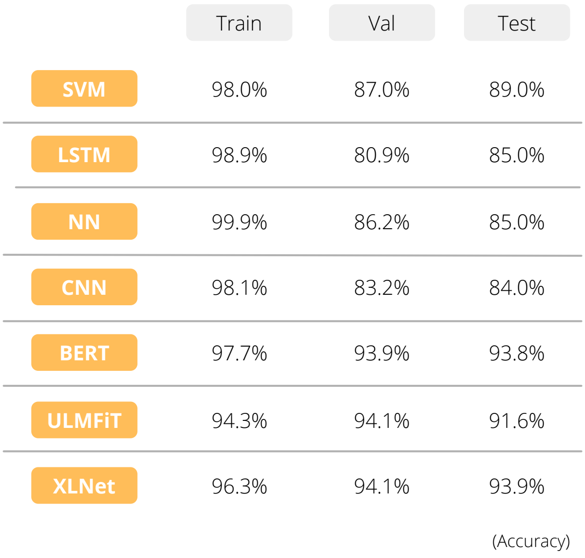
(Accuracy)

figure1.3

- ACL IMDb

  The ACL IMDb has 50000 data and is large enough for the model to train. Therefore the performance of the models is better than the one trained on UCI datasets. From the results in *Figure1.4*, we can see that SVM, LSTM, NN, and CNN have a serious over-fitting problem, where their test accuracy is almost 100%, but the test accuracy drops greatly.

Whereas transfer learning models such as BERT, ULMFiT, and XLNet perform well in the ACL IMDb dataset, which getting more than 90% accuracy.



	Train	Val	Test
SVM	98.0%	87.0%	89.0%
LSTM	98.9%	80.9%	85.0%
NN	99.9%	86.2%	85.0%
CNN	98.1%	83.2%	84.0%
BERT	97.7%	93.9%	93.8%
ULMFiT	94.3%	94.1%	91.6%
XLNet	96.3%	94.1%	93.9%

(Accuracy)

figure1.4

Conclusion

From our experiment, we can see that BERT and XLNet perform the best in terms of accuracy. This is likely to happen as they are the current state-of-the-art in NLP techniques. They can adapt to highly flexible data, and has acceptable accuracy. However, they are cost-consuming. For instance, when we fine-tune XLNet with only 3000 data and batch size of 4, it used up to 10 GB of GPU memory. In contrast, basic neural networks have other problems such as over-fitting of the model. Since sentences from different datasets have different structure and style, when we train a model with Amazon dataset and result in high accuracy, it might obtain poor accuracy when tested on IMDb dataset. It is hard to prevent model from over-fitting when the features of sentence are different in testing.

Reference

[1906.08237v2] XLNet: Generalized Autoregressive Pretraining for Language Understanding (arxiv.org)

[1801.06146] Universal Language Model Fine-tuning for Text Classification (arxiv.org)