Twitter Sentiment Analysis

Ioannis Ioannidis

University of Piraeus
NCSR Demokritos
Athens

24-06-2022

Outline

- Introduction
- Dataset Description
- Theoretical Background
- Experimental Setup
- Results
- Conclusions

Introduction

What is Sentiment Analysis?

- Sentiment analysis is a sub Machine Learning task where we want to determine which is the general sentiment of a given document.

Why?

- It is a really useful analysis since we could possibly determine the overall opinion about many domains.

How?

- Using Machine / Deep Learning techniques and Natural Language Processing we can extract the subjective information of a document and try to classify it according to its polarity such as positive or negative.

Problem?

- Sentiment analysis is actually far from to be solved since the language is very complex (objectivity/subjectivity, negation, vocabulary, grammar,...). However this is the main reason it constitutes such an interesting domain to work on.

Dataset Description

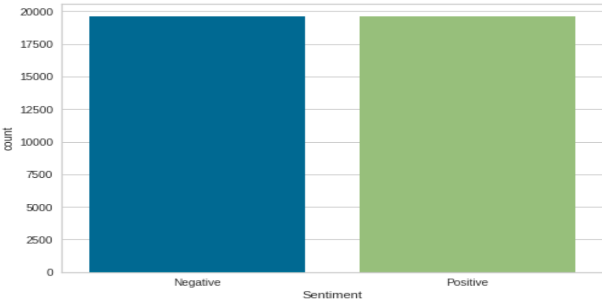
Options to gather data

- Some researchers select to built a program to collect automatically a corpus of tweets based on two classes, “positive” and “negative”, by querying Twitter with the respective emoticons.
- Others prefer to make their own dataset of tweets by collecting and annotating them manually (long and fastidious procedure).

Dataset Description

Our dataset

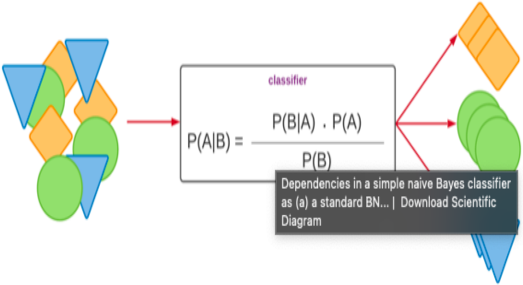
- In our case, we found an annotated twitter corpus on Kaggle https://www.kaggle.com.
- The corpus was quite large and unbalanced.
- In order to restore the balance in labels, as well as keep the training time in reasonable levels we kept a total of 40000 (20000 positive and 20000 negative) tweets for training.



Theoretical Background

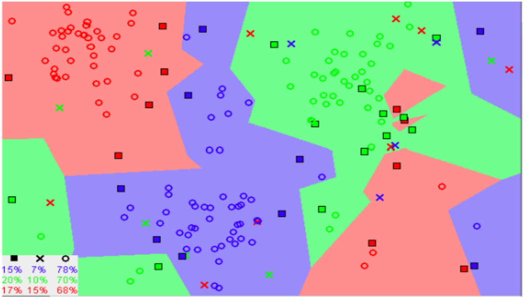
Naive Bayes

- Naïve Bayes classifier applies the Bayes theorem for probabilistic classification.
- By observing the input data of a given set of features or parameters Naïve Bayes classifier is able to calculate the probability of the input data belonging to a certain class.



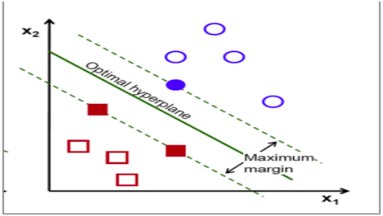
KNN

- KNN algorithm expresses the idea of similarity with some pretty easy mathematics
- It calculates the distance between points on a graph
- It assumes that similar things exist in close proximity
- In other words similar things are near to each other



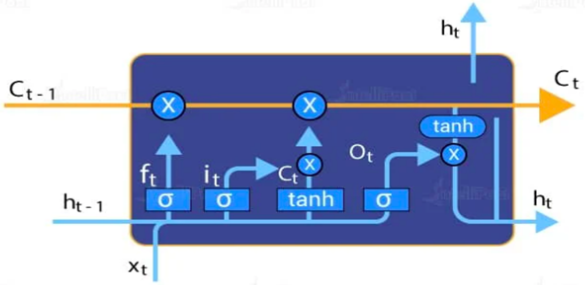
Support Vector Machines

- The objective of the support vector algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points
- To separate the two classes of data points there exist many possible hyperpanes
- SVM finds the one that has the maximum margin
- Maximizing the margin distance provides some reinforcement so that furure data points can be classifies with more confidence



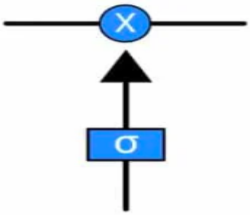
LSTM

- The central role of an LSTM model is held by a memory cell known as a 'cell state' that maintains its state over time.
- The cell state is the horizontal line that runs through the top of the below diagram.
- It can be visualized as a conveyor belt through which information just flows, unchanged.



LSTM

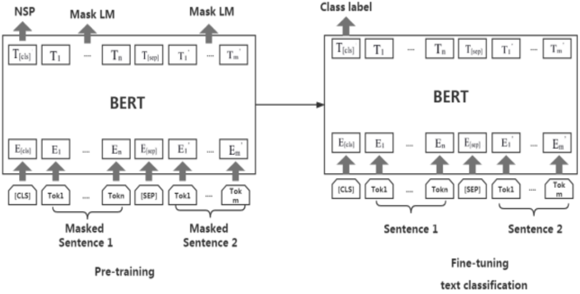
- Information can be added to or removed from the cell state in LSTM and is regulated by gates.
- These gates optionally let the information flow in and out of the cell. It contains a pointwise multiplication operation and a sigmoid neural net layer that assist the mechanism.
- The sigmoid layer gives out numbers between zero and one, where zero means "nothing should be let through" and one means "everything should be let through".



Theoretical Background

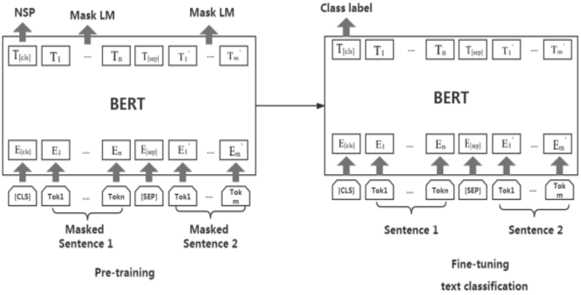
BERT Model

- BERT stands for Bidirectional Encoder Representations from Transformers.
- Bidirectional - To understand the text you're looking you'll have to look back (at the previous words) and forward (at the next words).



BERT Model

- Transformers - The Transformer reads entire sequences of tokens at once. The attention mechanism allows for learning contextual relations between words
- (Pre-trained) contextualized word embeddings - a way to encode words based on their meaning/context



Data Preprocessing

Before passing tweets through models we apply a clean-up function which:

- turns all letters to lowercase
- removes punctuation
- removes stopwords
- removes links
- removes emojis
- stems each word

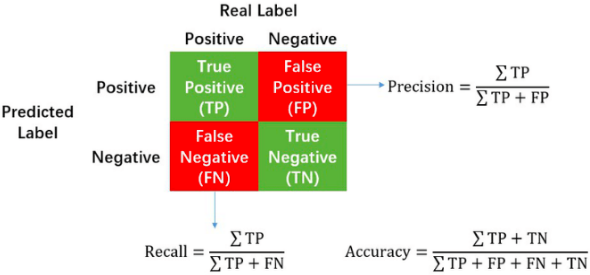
2	sir may good pick from the beginning disappoin...	Negative

↓

2	sir may good pick begin disappoint toward way ...	Negative

Evaluation Metrics

- *Precision* is the number of True Positives (TP) divided by the total number of elements labeled as belonging to the positive class.
- *Recall* is the number of True Positives (TP) divided by the total number of elements that actually belong to the positive class.



Experimental Setup

Evaluation Metrics

- There is a trade-off between precision and recall, where increasing one decreases the other.
- We usually use measures that combine precision and recall, such as F1-score.
- F1-score can be interpreted as the weighted average of the precision and recall and its a good measure of a model's accuracy.

$$F1 = \frac{2 \times (precision \times recall)}{precision + recall}$$

Results

Final Comparison

- We trained the models on 40000 Tweets (train set).
- We evaluated the models on 35000 Tweets (validation set).
- All models got the exact same train and validation sets.

	Accuracy	Precision	Recall	F1-Score
Naive Bayes	68.53%	0.73	0.59	0.65
KNN	62.92%	0.6	0.79	0.68
SVM	80.33%	0.83	0.76	0.79
LSTM	76.15%	0.81	0.68	0.74
BERT	84.52%	0.85	0.83	0.84

Table 1: Results

Conclusion

- Sentiment Analysis is a hot topic in Machine Learning.
- We are still far to detect the sentiments of a corpus of texts very accurately because of the complexity in the English language.
- All results were quite good, considering the large amount of validation data.
- We could further improve our classifier by trying to extract more features from the tweets, trying different kinds of features, tuning the hyperparameters, or combine models.