Twitter Sentiment Analysis

Presented by: Yu Xiao

Introduction

The goal of this project is to classify binary sentiments (either positive or negative) of tweets. This is a supervised, binary classification task. I experimented with various models (Neural Networks and others) and evaluated their performances.

Dataset

Sentiment140 is a labelled dataset that contains 1.6 million tweets collecting from Twitter API. Following 2 fields were used in my projects:

● target: the polarity of the tweet
(0 = negative, 2 = neutral, 4 = positive)
● text: the text of the tweet

Data Cleaning:

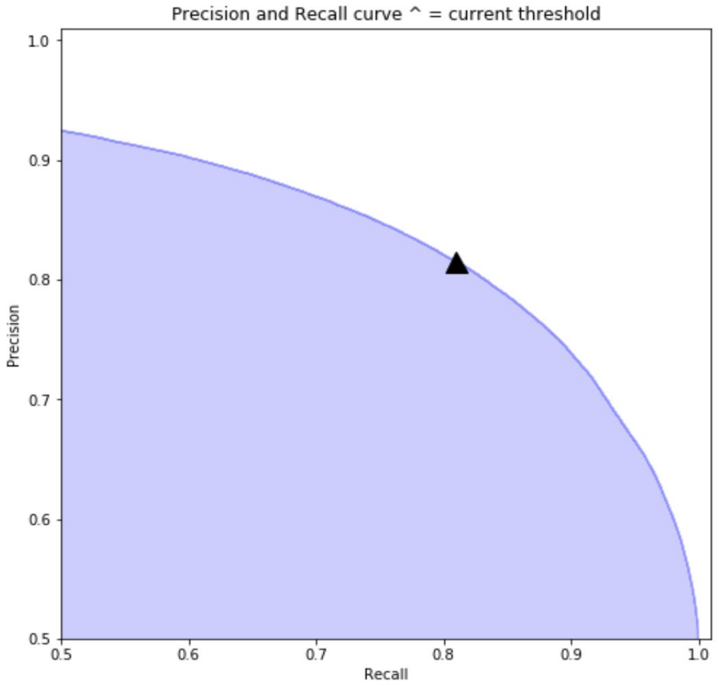
- Removed urls, @username, #topic
- Tokenized and lemmatized tweets using nltk library

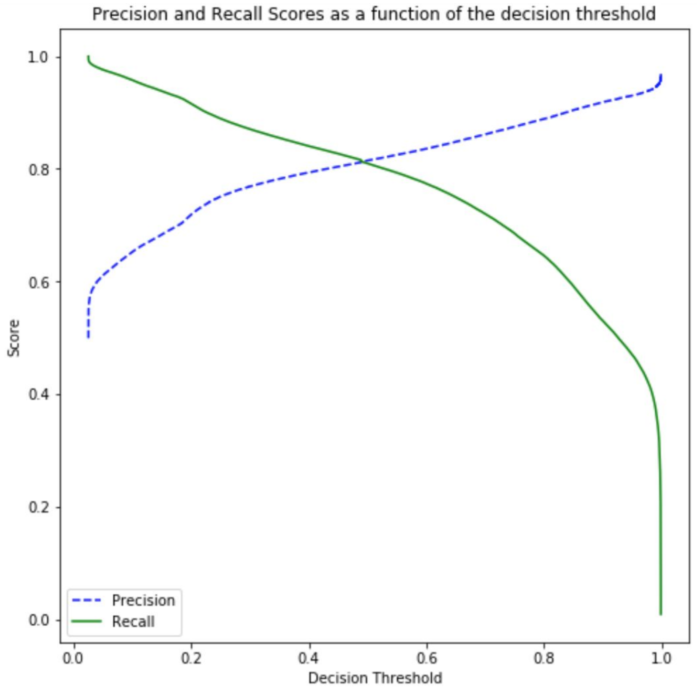
Methods

- Trained a Support Vector Machine(SVM) classifier, a Naive Bayes classifier and a Logistic Regression(LR) classifier on TF-IDF weighted word frequency features. [Sklearn]
- Trained a customized 5 layers CNN model (1 embedding layer and 4 convolutional layers). [Keras]
- Trained a customized 3 layers LSTM model (1 embedding layer and 2 LSTM layers). [Keras]
- Trained the combination of CNN and LSTM (1 embedding layer, 4 convolutional layers followed by 2 LSTM layers). [Keras]
- Trained a word2vec model follow by fully connected layers. [Gensim & Keras]

Evaluation Metric

Weight F1 score: 2 * (precision * recall) / (precision + recall)





Results

Model	With Tokenization & Lemmatization	Without Tokenization or Lemmatization
Naive Bayes	0.77647	0.77650
Logistic Regression	0.79042	0.79185
SVM	0.78098	0.78367

Chose TF-IDF vectorizer to transfer cleaned tweets to feature vectors

Neural Nets Model	With Tokenization & Lemmatization
CNN	0.81422
LSTM	0.81643
CNN+LSTM	0.81369
Word2Vec	0.79557

Analysis

- SVM tries to find the widest possible separating margin, while LG optimizes the log likelihood function. Both LG and SVM are commonly used for binary classification. They are better than Naive Bayes which assumes the features are conditional independent.
- Word2Vec seems like a better Vectorizer than TF-IDF (importance).
- All Neural Nets models outperformed other 3 models. LSTM model achieved best F1 score as it could capture the sequential information. CNN learned weights through different combination of tokens (with local connectivity). The combination of CNN and LSTM also achieved a competitive result as it combined the strength of two, but it didn’t outperform CNN or LSTM alone because CNN might lose some sequential information and thus LSTM didn’t work as expected.