Sentiment Analysis of IMDb movie reviews

Group 13 : Anirudh Kumar Agrawal (11098)
Anjani Kumar (11101)
Nitin Kumar Singh (11472)
Sangharsh Aglave (11643)

Problem Statement

Given a movie review, classify it as having ‘positive’ or ‘negative’ sentiment.

Training Set: 25000 supervised data + 50000 unsupervised data

Preprocessing

- Remove the HTML tags to get original text
- Remove Punctuation marks and any other non-alphabetic symbols
- Convert all data to lower case

Word2Vec

- An unsupervised method for creating a vector representation for each word in the vocabulary
- It retains the relationship of the words in the vector space
- vec[queen] – vec[king] = vec[woman] - vec [man]

Word2Vec Training

- It uses two training methods
  - Continuous Bag of Words :- In the CBOW the goal is to predict a word given surrounding words.

  - Skip Gram model with negative sampling :- This is converse to CBOW model. The goal is to predict a window of words given a single word.

Doc2Vec

This method is similar to the Word2Vec, except we use a paragraph vector along with the word vectors for classification . There are two method of training a Doc2Vec

i) Distributed Memory.

ii) Distributed Bag of Words.

Doc2Vec

Distributed Memory:- Distributed Memory predict the words given its previous words and paragraph vector. Stochastic gradient descent is used to train it and the gradient is obtained via backpropogation.

At prediction time we have to perform a inference to compute new paragraph vector. This is done by gradient descent. Other parameters of the model are kept fixed at this step

Distributed Bag of Words:- Predict a random group of words in a paragraph given only its pararaph vector.

Bag of Centroids

- Number of clusters = |Vocalbury Size|/5
- Performs clustering on the Vocalbury using the word vectors from word2vec
- For each review create a bag of centroids by calculating the number of times for each cluster a word occurs in that review

Tfidf

Tfidf is a measure of how important a word is to a document.

tf - term frequency in a document

idf - inverse document frequency,

log of no. of documents divided by the number of documents the term is present.

tfidf = tf*idf

Classification Approaches

- SVM
- RandomForest
- Naive Bayesian
- Adaboost
- Logistic Regression

Results

	Tfidf	Word2Vec	Para2Vec
Random Forest	0.79	0.84	0.77
SVM	0.81	0.87	0.85
Adaboost	0.79	0.84	0.81
Logistic Regression	0.95	0.91	0.94
Naive Bayes	0.78	0.77	0.79

* All values reported are AUC

Further Improvements

- Word2Vec could not be trained on more data (due to hardware constraints), which should give better results.