Machine Learning Engineer Nanodegree
Capstone Proposal
Pushpankar Kumar Pushp
February 26th, 2017

Proposal

Domain Background

There is huge amount of text on the web. Most of the information is in the form of text. Text is also the most commonly used method for tele-communication. Organizing and classifying these texts is more important than ever. It can be used to find interesting article for a user, discard irrelevant information, and word sense disambiguation. Doing this by hand is very time consuming and difficult. Since early 90's machine learning has gained popularity for this task. I will be comparing traditional machine learning approach and deep learning for text classification.

A comparision of traditional machine learing approaches has been done by Bo Pang and Lillian Lee et al.[1]. Andrew L. Maas et al. has used supervised and unsupervised learning techniques for sentiment analysis in his paper[2].

Classifying text is the first step towards building AI agents that can properly communicate with humans.

Problem Statement

I will be analysing the sentiment of a text.This is a binary classification task. These text can be of arbitrary length. Sentiment of a text can either be positive or negative. The sentiment of the text can change dramatically with a single word. Unlike text categorization, most of the sentiment information lies in frequently used words.

Datasets and Inputs

I will be using IMDB movie review dataset for sentiment analysis. This dataset is very good fit for the this task as it contains a lot of labeled examples and unlabeled examples as well. This dataset contains a total of 100,000 multi-paragraph reviews. The labels of labels of labeled examples are in binary format, 0 for negative sentiment and 1 for positive sentiment.

This dataset is suitable for my sentiment analysis task as in both cases we need to classify text as positive or negative. I will use this dataset to train model and then test them on my own text.

I got this dataset from Kaggle.com[3].

Solution Statement

The sentiment of a text can be predicted using the sentiment of words it contain. For computing the sentiment of words either count based method or word2vec can be used. Count based method works by computing the probability of occurance of a word in a class. Word2vec method works by finding similar words and properly organizing them. Then these word vectors can be fed to recurrent neural network for classifying text. These word vector can also be used for computing average vector of a review and training algorithms like random forest on it.

Benchmark Model

I will be using naive bayes classifier as my benchmark. I will be comparing the accuracy of naive bayes with the accuracy of other techniques.

Evaluation Metrics

I will be using accuracy, precision, recall and f1 scores as my metric. Accuracy is percentage of correct prediction out of total predictions. For every text I will predict both the probability of being positive and negative. Then I will finally predict the sentiment whose probability is more.
Then finally I will compare different machine learning techniques like naive bayes, random forest on word vectors and natural language processing for this task.

Project Design

The data has some HTML tags. I will remove these tags. Reviews also contains numbers which does not usually affect the sentiment of text. So, I will remove them too. I will not be considering special symbols to manage the complexity of the algorithm although they can affect the sentiment. I think plain text can be used to predict the sentiment with good accuracy.

I will then use naive bayes for sentiment analysis to get the base line. For naive bayes, I will first create a bag of words which is the count of words in each class. Then I will compute sentiment of words probabilistically using the count based method. Finally I will apply naive bayes for predicting the sentiment of the text. Stop words like a, the, with etc. might not add much value to our task. So I will compare model’s accuracy in both the cases: keeping the stopwords and removing them.

In next step, I will create a vector representation of the words. I will use skip-gram model for this task. Then I will use some traditional machine learning approach for sentiment analysis like KNN. In word2vec, I will not be removing the stop words because Deep learning models require more data than traditional methods. I will use this word embedding for other prediction models as well.

Then, I will predict sentiments using natural language processing techniques. I will use recurrent neural network for this and LSTM cell as it does not have vanishing gradient descent problem. LSTM is necessary because the text can be a long sequence of words which might create vanishing gradient descent problem in vanilla recurrent neural network architecture.

Then finally I will compare these different models based on their accuracy.

----

1 – Thumbs up? Sentiment Classification using machine learning Techniques.
http://www.aclweb.org/anthology/W02-1011
2 – Learning word vectors for sentiment analysis.
3 - Data Source:- https://www.kaggle.com/c/word2vec-nlp-tutorial/data