Capstone proposal

Zoe Zhou

November 29th, 2017

Domain Background

NLP is the acronym of Natural Language Processing. Using Machine Learning in this area, computers can "read" articles, extract useful information from unstructured Data, classify news, "understand" meanings and calculate words similarity, and even more, computers are capable of judging the sentiment of reviews.

Generally, machine learning algorithms require the input to be represented as a fixed-length feature vector. When it comes to texts, one of the most common fixed-length features is bag-of-words. BoW is a simple but applicable method in classifying articles, such as spam emails detection. Despite its popularity, BoW pays heavy attention to words' frequency and has two major weaknesses: ignores the ordering of words and the meaning of words. These weaknesses has greatly limited BoW’s competence in text mining.

In 2013, Google published Word2vec method, a deep learning achievement in NLP, which assigned computers new abilities to understand words internal and external meanings. To everybody's surprise, without learning any special semantic or syntax rules, just by inputing lots of articles to a shallow-layered neural network model, computers could know words' intension so well that they can even achieve this famous equation: "king - man + woman = queen".

On 16 May 2014 , Quoc V. Le, Tomas Mikolov published the famous article “Distributed Representations of Sentences and Documents”, in which they developed an efficient way inspired by word2vec to represent features of paragraphs in a dense fixed-length vector. That is the origin of today’s Doc2vec algorithm. By using Doc2vec algorithm, engineers could gain efficient text features and achieve a new state-of-art in text classification and sentiment analyzing.

- https://rare-technologies.com/word2vec-tutorial/
- https://cs.stanford.edu/~quocle/paragraph_vector.pdf

Problem Statement

In this capstone project, I will dive into an interesting competition on Kaggle, "bag of words meet bag of popconrn", you can click on the title to read more details about the competition .

The goal of this competition is to predict sentiment of all the movie reviews in test data.

Datasets and Inputs

The dataset for this project can be found in the data section of this competition on the kaggle platform.

Data Set

The labeled data set consists of 50,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of 0, and rating >=7 have a sentiment score of 1. No individual movie has more than 30 reviews. The 25,000 review labeled training set does not include any of the same movies as the 25,000 review test set. In addition, there are another 50,000 IMDB reviews provided without any rating labels.

File descriptions

- labeledTrainData - The labeled training set. The file is tab-delimited and has a header row followed by 25,000 rows containing an id, sentiment, and text for each review.
- testData - The test set. The tab-delimited file has a header row followed by 25,000 rows containing an id and text for each review. Your task is to predict the sentiment for each one.

- unlabeledTrainData - An extra training set with no labels. The tab-delimited file has a header row followed by 50,000 rows containing an id and text for each review.
- sampleSubmission - A comma-delimited sample submission file in the correct format.

Data fields

- id - Unique ID of each review
- sentiment - Sentiment of the review; 1 for positive reviews and 0 for negative reviews
- review - Text of the review

Solution Statement

To get a better result, I will use Doc2vec algorithm to extract and represent features of movie reviews. There're different models using different parameters, such as PV-DMC (paragraph2vec distributed memory concatenation), PV-DMM(paragraph2vec distributed memory average), PV-DBOW(paragraph2vec distributed bag of words), and so on. I'd like to try each of them to select the best one.

Because Doc2vec is an unsupervised learning algorithm, we can use unlabeled training reviews and even the test reviews to gain paragraph2vec vectors for movie reviews. Using this consideration, we could maximize the corpus to better represent our text features.

Besides, I will try and compare multiple classification algorithms,such as random forest, logic regression, naïve bayes, support vector machine, ensemble, neural network, to see in text sentiment classification scenario, which classification algorithm predict the best coping with doc2vec feature vectors. I believe the whole process of this capstone project will be full of challenge and joy.

Benchmark Model

This competion's benchmark model is a "Word2Vec - Bag of Centroids" model, which has an AUC of 0.84528, ranking after the No417.

Evaluation Metrics

Submissions are judged on area under the ROC(**receiver operating characteristic curve**) curve.

Related metrics are :

true positive rate (TPR,sensitivity, recall)

false positive rate (FPR, 1- specificity)

Project Design

The main workflow of this project is as following :

- import all reviews
- normalize text into words and paragraphs
- train paragraph2vec using Doc2vec
- tune train models and parameters
- calculate error rate and compare models
- choose the best Doc2vec model to infer features of test movie reviews
- try and compare different classification algorithms with feature vectors
- predict and upload result to kaggle
- get feedback from kaggle and tune model
- summary of achievements and potential further improvements