Proposal

Movie review classification with machine learning techniques into positive or negative.

Domain Background

With the amount of opinions expressed on social media, a sense of public opinions can be assessed if the “polarity” of opinions on a specific topic can be assessed. An accurate assessment of public opinion is a very useful tool for marketing.

Unfortunately, this is not a trivial tasks given the scale of data available and variation in the sentence construction. Since human languages analysis with hard coded rules will be brittle, machine learning is the natural answer. They can be potentially trained to be able to classify the statements according to the inferred sentiment.

Sentiment analysis is a challenging subject in machine learning. People express their emotions in language that is often obscured by sarcasm, ambiguity, and plays on words, all of which could be very misleading for both humans and computers.

I work in semantics development for a graphical modelling language in my day job. This has increased my interest in natural language processing (and semantic inference in particular).

I used a kaggle competition https://www.kaggle.com/c/word2vec-nlp-tutorial as reference for this. The introductory section gave the traditional approach to Semantic classification of sentences and the Deep Learning inspired approach to understanding word semantics.

I also used the following as reference for understanding about the concept of paragraph vector for taking word ordering also into context
http://cs.stanford.edu/~quocle/paragraph_vector.pdf
https://districtdatalabs.silvrback.com/modern-methods-for-sentiment-analysis

In addition to this there is some interesting parsing and deep learning based research going on at Stanford
http://nlp.stanford.edu/sentiment/

Problem Statement

The primary objective of my capstone project is to classify movie reviews into two classes - positive or negative. I will use a few different techniques and comparing their results.

For this I will use Kaggle project https://www.kaggle.com/c/word2vec-nlp-tutorial as a reference and data source. I will also explore the new approaches used by Stanford NLP team (http://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf) if possible.

Datasets and Inputs

I shall draw the dataset(s) from https://www.kaggle.com/c/word2vec-nlp-tutorial/data

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.

The input data consists of the following data repository. There is an option to try unsupervised learning technique also and therefore there are both labeled and unlabeled data sets.

- **Labeled Train Data** - 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.
- **Test Data** - 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.
- **Unlabeled Train Data** - 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.
- **Sample Submission** - A comma-delimited sample submission file in the correct format.

Solution Statement

I shall implement a traditional bag of words based classifier for sentiment analysis to set the benchmark for this. I will use different recent approaches for sentiment analysis:

1. Word vector with averaging (weighted and otherwise)
2. Paragraph vector
3. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

I shall measure the performance over the evaluation metric (ROC curve area) on test data. In addition to that there will also be the additional challenge of whether the data will need to be fully supervised or not (paragraph vector does not need to be fully supervised as per the quoted literature and needs no parsing).

The solution therefore will be to identify the optimal approach among the algorithms identified to use for classifying movie reviews.

Benchmark Model

The traditional bag of words combined with a Random Forest shall serve as the benchmark model. The area under ROC curve will serve as a metric to compare with the other approaches.

Evaluation Metrics

Area under receiver operating characteristic (**ROC**) curve and f1 score of the mechanism. Accuracy is measured by the area under the ROC curve. An area of 1 represents a perfect test. On the other hand, an area of .5 represents a worthless test.

ROC curve would plot the False positive rate (x-axis) v/s True positive rate (y-axis). Intuitively speaking, greater area under curve would mean higher true positives to false positive ratio.

Project Design

Historically, Bag of Words has been used to do a sentiment analysis. But they fail to capture the sentiment information that is central to many word meanings and important for a wide range of NLP tasks. This is where word vector usage should outperform bag of words model.

I would keep a common pre-processing API as follows

1. Read the 25,000 reviews
2. Remove the HTML tags from the reviews
3. (optionally) Remove punctuation marks and stop words (https://en.wikipedia.org/wiki/Stop_words) to reduce feature vector dimensions.
4. Tokenize the processed reviews since each word is a discrete entity.

Traditional Approach

With Bag of Words approach we can classify this processed data using random forests which can give us a reasonable benchmark to beat.

Word Vector Approach

As an alternative to Bag of Words, I will try Word2Vec (https://code.google.com/archive/p/word2vec/). It is essentially a neural network approach to learn distributed representations of words. The claim is that it shall be easy to train it faster than recurrent neural network architectures.

Word2Vec is an unsupervised learning algorithm. It takes a text corpus as input and produces the word vectors as output. It first constructs a vocabulary from the training text data and then learns vector representation of words. The resulting word vector file can be used as a feature in our application.

With the semantic relationship mapping, we can use this to cluster the words with similar semantic meaning together. Beyond this, we shall have bag of centroid (of clusters) instead of bag of words. This should significantly reduce the dimensions of the feature vector over the “word space”. Beyond this we could use a similar classifier as a Random Forest.

Paragraph Vector Approach

To further enhance the performance, I will try the usage of Paragraph Vector (https://cs.stanford.edu/~quocle/paragraph_vector.pdf) which shall take into account the ordering of the words (unlike bag of words) similar to a n-gram model with very large n.

Recursive Neural Tensor Network

I want to explore this if time permits. It is an interesting approach. However, unlike other approaches it relies on parsing to capture word order.
(http://nlp.stanford.edu/~socherr/EMNLP2013_RNTN.pdf)