# Machine Learning Engineer Nanodegree Capstone Project

Agost Biro

July 11, 2017

# I. Definition

## Project Overview

The project presented here involves the automated prediction of movie review sentiments from IMDB based on the Word2vec word embedding model. Sentiment analysis is one of the fundamental tasks of natural language processing. It is commonly applied to process customer reviews, implement recommender systems and to understand sentiments in social media. SemEval (short for semantic evaluation) is an annual competition designed to facilitate and track the improvements of semantic analysis systems. The best submissions to the SemEval 2016 Sentiment analysis in Twitter task overwhelmingly relied on word embeddings, the most popular of which was Word2vec. (Nakov et al., 2016)

Vector embeddings are learned vector representations of information such that the constructed vector space exhibits regularities useful for a specific task. Word embeddings are a natural language processing technique where a vector embedding is created from words or phrases. The purpose of these embeddings is to provide a representation of the input that is convenient to operate upon for other natural language processing or information retrieval methods. (Wikipedia: Word embedding) The Word2vec model is popular because it offers state-of-the-art performance in evaluating semantic and syntactic similarities between words with relatively low training times. (Mikolov et al., 2013) I find the topic of vector embeddings interesting, because I suspect that it might be possible to discover new information about a corpus based on the analysis of its embeddings.

## Problem Statement

The aim of this project is to examine whether feature vectors created with the Word2vec word embedding model perform well on the Bag of Words Meets Bags of Popcorn Kaggle competition. The task involves the binary classification of movie reviews from IMDB as expressing either positive or negative sentiment.

The solution to this problem is to provide a binary sentiment classification (0 for negative, 1 for positive) for each review in the testing data with a logistic regression classifier using feature

vectors from a self-trained Word2vec model. The solutions are uploaded to Kaggle that scores them. The best solution achieves the highest Kaggle-score.

## Metrics

Kaggle uses the area under the receiver operating characteristic (ROC) curve to evaluate submissions. An ROC curve is produced by plotting the true positive rate (TPR) against the false positive rate (FPR) at various discrimination thresholds [0, 1] and connecting the points. TPR = TP / (TP + FN) and FPR = FP / (FP + TN), where TP is the number of true positives, FN is the number of false negatives, FP is the number of false positives and TN is the number of true negatives. (Wikipedia: Receiver operating characteristic)

Using the area under the ROC curve as an evaluation metric in this case is somewhat confusing, as the submission should only contain a binary classification (ie. 0 or 1) and not the probability of a review belonging to one class or the other. So in this case, only one point will be plotted after calculating the TPR and FPR on the submission with an unknown threshold value from the evaluator's point of view. The curve is then completed by connecting the points (0, 0), (TPR, FPR) and (1, 1). (Source: Kaggle forums, Ilya Ezepov and franck)

# II. Analysis

## Data Exploration

The dataset is provided courtesy of Kaggle and it was collected in association with (Maas et al, 2011.):

> 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. (Chapman, 2014)

The following is a randomly selected review from the unlabeled dataset:

> If you enjoy suspending disbelief and you appreciated \"Gremlins\", then this new offering form executive producer Steven Spielberg and director Joe Dante is guaranteed to keep you thoroughly entertained for over two hours (though you won't believe you've been sitting down that long). This extremely lively comedy concerns the misadventures of inept supermarket assistant manager Jack Putter (Martin Short), who is accidentally injected with a microscopic pod

> which contains a crack pilot who was supposed to be exploring a rabbit. So while Jack is beside himself with fears of demonic possession (he's hearing the voice of pilot Tuck Pendleton - Dennis Quaid), he is being pursued by government agencies and ruthless criminals. It all might sound rather fantastic, and it is, in the best possible way! Comic genius Martin Short really makes this movie his own with some outrageous antics (not unlike Jerry Lewis) that carry scene after scene, hilariously. Second billing really should go to Kevin McCarthy (though his is a small role) for his eccentric portrayal of Scrimshaw, the megalomaniac criminal who is 'just in it for the money'. Also great is Wendy Schaal, a checkout chick who is a little unsure about whether she should be dating the erratic Jack. Dennis Quaid makes a rugged, handsome hero while his real life wife Meg Ryan is a pretty heroine, in one of her better roles. Other strong support comes from Fiona Lewis and Henry Gibson.<br /><br />Director Dante and screenwriters Chip Proser and Jeffrey Boam (Proser wrote the original story) have a ball with the wild premise (based loosely on \"Fantastic Voyage\" - 1966) and milk the comedy aspect dry, whilst the Academy Award winning visual effects team recreate the inside of the human body spectacularly, and Jerry Goldsmith provides a very effective and at times rousing score.<br /><br />\"Innerspace\" is one of those movies that's loads of fun no matter how many times you see it! Several scenes are still riotously funny even for the fifth time. If only for Martin Short's manic performance and the brilliant special f/x, this picture is a must.<br /><br />NB What looked like an opening for a definite sequel never did eventuate, or at least no yet!<br /><br />Saturday, July 15, 1995 - Video

The labeled dataset was split into two parts to create a training and a validation dataset with a ratio of 5:1 (ie. 20% of the data was used for validation). The unlabeled training data was used to train the Word2vec model. The labeled dataset is balanced in that it contains the same number of positive and negative reviews and there is no correlation between the number of words in a review and the sentiment. The reviews have to be preprocessed to be cleaned of html tags and symbols.

## Exploratory Visualization

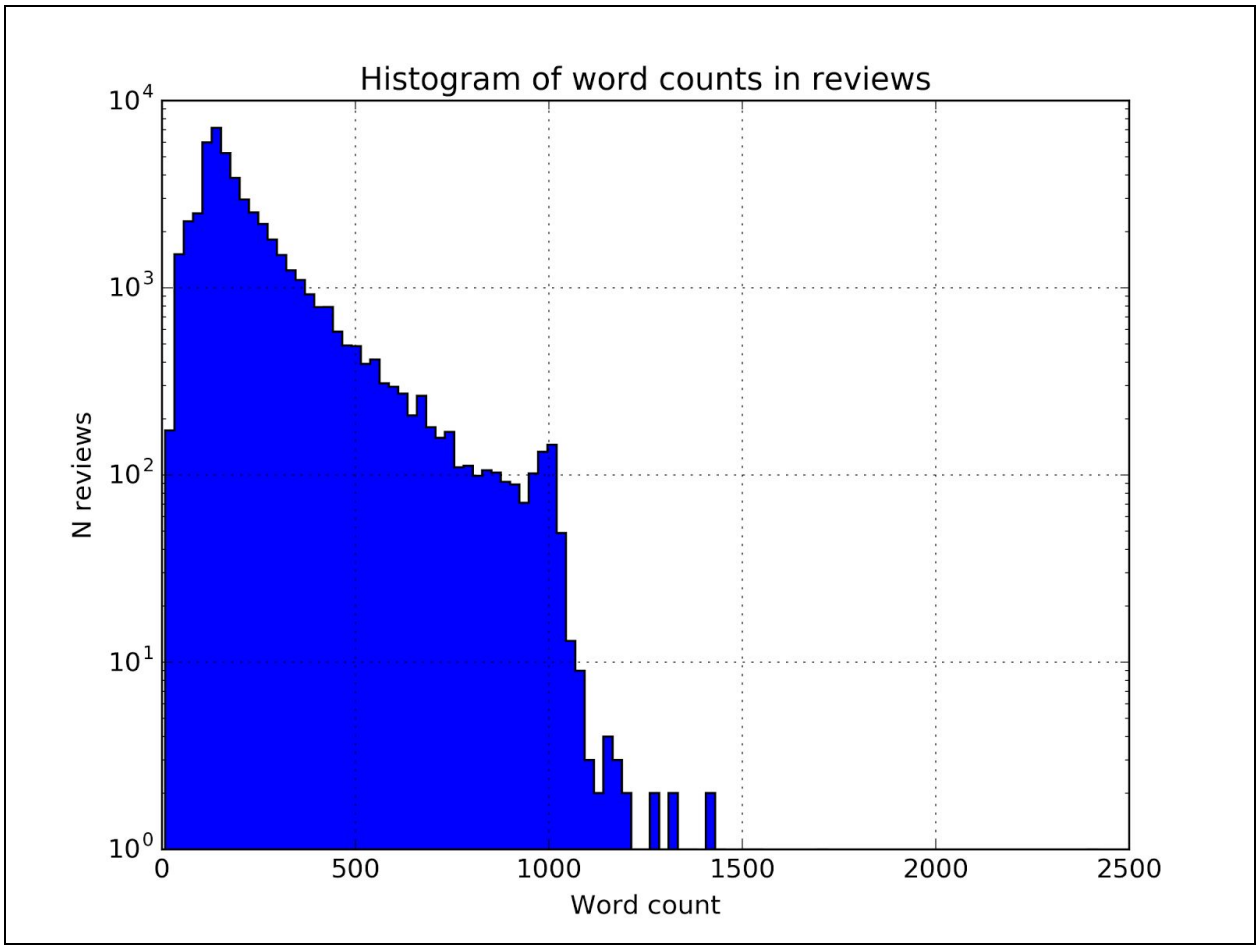


Fig. 1. Histogram of word counts in reviews

Fig. 1. displays the distribution of the length of the reviews using a logarithmic scale on the y-axis. We can conclude based on the graph that the review lengths roughly follow a power law distribution with a sharp cutoff around the 1000 mark.

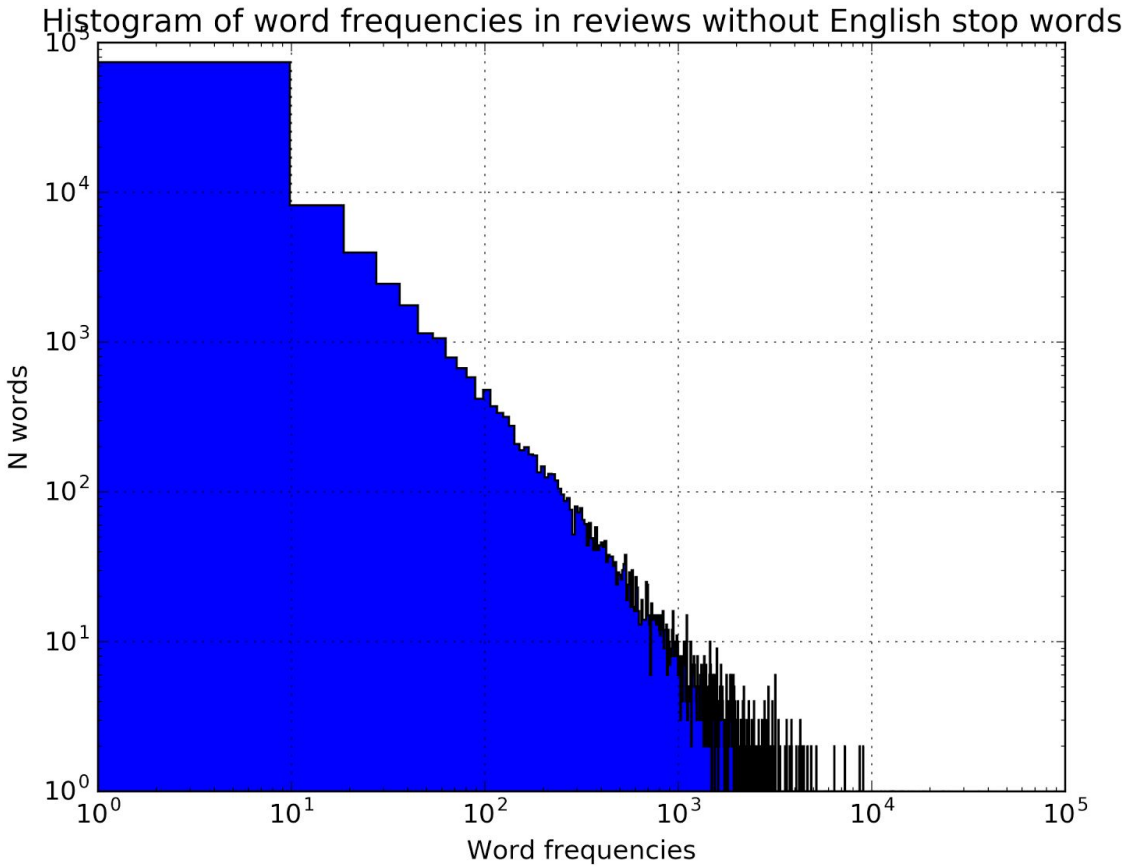


Fig. 2. Histogram of word frequencies in reviews without English stopwords

Fig. 2. displays the distribution of word frequencies in the unlabeled dataset on a logarithmic scale along both axes. The vocabulary size excluding stopwords is 101,953. The important takeaway from this figure is that the word frequencies follow a power law distribution and that ~90% of the words occur only 10 times or fewer in the corpus. Dealing with such a rare words is hard for the Word2vec model.

The 10 most frequently used words (excluding English stop words) in the unlabeled dataset are:

| **Word** | **Frequency** |
|---|---|
| movie | 88154 |
| film | 81072 |
| like | 40700 |
| just | 35807 |

| | |
|---|---|
| good | 29724 |
| time | 25111 |
| story | 23581 |
| really | 23187 |
| bad | 19056 |
| people | 18591 |

# Algorithms and Techniques

## Logistic Regression

Logistic regression is a supervised binary classification technique. Conceptually, logistic regression starts with performing a linear regression parameterized by Θ on the covariate X and then squashes its output with the logistic function σ, the result of which can be interpreted as the conditional probability of the response variable Y belonging to class 1 given the covariate X (Goodfellow et al., 2016, p. 137.):

$$p(y = 1 \mid \mathcal{x}; \Theta) = \sigma(\Theta^T \mathcal{x}) = e^{\Theta T \mathcal{x}} / (1 + e^{\Theta T \mathcal{x}}).$$

A convenient characteristic of the logistic function is that,

$$\Theta^T \mathcal{x} = \ln(p / (1 - p)),$$

thus $\Theta^T \mathcal{x}$ can be interpreted as the natural logarithm of the odds that Y belongs to class 1. The parameter Θ can be estimated by performing maximum likelihood estimation on the training data by interpreting $\sigma(\Theta^T \mathcal{x})$ as a probability density function. (Wasserman, 2004, p. 223-224.)

Logistic regression is fast in training and inference and despite its simplicity, performs well in a wide variety of cases. A common challenge associated with logistic regression is that maximum likelihood estimation can badly overfit for linearly separable datasets. This can be avoided by using regularization in the error function. (Bishop, 2006, p. 206.)

## Word2vec

Word2vec is a feedforward neural network language model (NNLM) that produces word embeddings. NNLMs were designed to produce dense vector representations of words such that similarities between them are conserved without losing the ability to distinguish between similar words. (Goodfellow et al., 2016, p. 451.) A classical NNLM consists of an input layer, where words are encoded in a one-hot representation, a projection layer that condenses the input vectors, a hidden layer that computes the probability distribution over all the words in the vocabulary and an output layer. Word2vec improves on a classical NNLM by removing the

hidden layer to trade representational power for efficiency and thus the ability to train using a lot more data which generally gives better results in machine learning. (Mikolov et al., 2013)

The first variant of Word2vec, called Continuous Bag-of-Words Model (CBOW), tries to predict the current word based on its context from a number of other words. This model can not distinguish between different positions in the context as they are all projected to the same position. The second variant, called Continuous Skip-gram Model, tries to predict the context from the current word with words farther away being undersampled to give them less importance. During training, the models have to predict the current word or its context from a small number of examples with logistic regression instead of trying to discriminate among all the words in the corpus. The Skip-gram model works much better on semantic similarity tasks than the CBOW model. (Mikolov et al., 2013) The embeddings are found in the weights between the input layer and the output layer that is a V × N matrix where V is the vocabulary size and N is the number of embedding dimensions. (Rong, 2014)

## Benchmark

The benchmark is a cross-validated logistic regression model trained using feature vectors produced from the 5000 most frequent words (excluding English stop words) with the bag-of-words method. The logistic regression model was chosen because it performed best on the validation dataset among a decision tree classifier, a random forest classifier and a support vector classifier. The model received a 0.8687 score from Kaggle for its predictions on the testing dataset.

# III. Methodology

## Data Preprocessing

The training data is found in tab-separated CSV files. Following the Kaggle tutorial, the dataset was parsed with the Pandas library and the HTML markup was removed with the Beautfil Soup library. A list of lowercase words were produced from each review and punctuation was removed with standard string manipulation tools. In addition to what is suggested by Kaggle here, a validation set was created by splitting the training data 80:20. Source file: sentiment_analysis/make_dataset.py

## Implementation

### Bag-of-words feature vectors

The bag-of-words feature vectors were created using the cleaned unlabeled dataset with the help of scikit-learn library's CountVectorizer with 5000 max features parameter and English stop

word removal. Three term matrices were produced for the training, validation and testing datasets where rows contains the vectors and each row corresponds to a row in their respective dataset. The term matrices are saved as pickle files. Source file: sentiment_analysis/make_bag_of_words_features.py

### Word2vec model and feature vectors

The neural network that trains the Word2vec model was implemented with TensorFlow as a fork of the optimized Word2vec implementation from the TensorFlow models repository. The Word2vec model is using the skipgram model with stochastic gradient descent for training. The embedding vectors have a dimensionality of 200 and the number of epochs to train is set to 15 with a batch size of 500 and a learning rate of 0.025. The model uses a window size of 5 and words that appear with a high frequency (over 1e-4) are randomly removed from the window (subsampling). Only words that have 5 or more occurrences are included in the vocabulary. The feature vectors are produced by taking the mean of the normalized embedding vectors of all the words in a review. The model trained with these parameters is referred to as the default Word2vec model in the following. Source file: sentiment_analysis/word2vec/word2vec.py

### Binary classification models

The following four binary classification models are implemented using the scikit-learn library with the intention of quickly checking their performance on the validation set: LogisticRegressionCV, DecisionTreeClassifier, RandomForestClassifier, SVC (support vector classifier). The metric used is the area under the receiver operating characteristic as explained in the first section. These models also produce predictions on the testing set. Source files are in the following submodule: sentiment_analysis/models.

## Refinement

Three unsuccessful attempts were made to improve the initial model:

- Tried training Word2vec without subsampling (ie. not undersampling frequent words). Evaluation ROC AUC score on the validation dataset decreased from 0.8923 on the default model to 0.8904.
- Tried training Word2vec on the larger text8 corpus that has 17,005,207 words versus the 11,876,820 words in the unlabeled dataset. Evaluation ROC AUC score on the validation dataset decreased from 0.8923 on the default model to 0.872.
- Tried clustering the word embeddings from the default training with affinity propagation and then creating feature vectors by counting the number of words in a review that belong to each cluster. Eg. if two words in a review belonged to cluster 1 and one word in a review belonged to cluster 2, the feature vector would be: [2, 1]. Evaluation ROC AUC score on the validation decreased from 0.8923 on the default model to 0.8804.

# IV. Results

## Model Evaluation and Validation

The cross-validated logistic regression model produced the highest scores on the validation set using feature vectors from default Word2vec model, so predictions from this model were submitted to Kaggle and received a 0.8204 score. The model can be trusted to generalize well, as it has been evaluated by Kaggle on a random subset of the testing data whose labels are unknown. As a sanity check, the procedure was repeated with the word embeddings in the default Word2vec replaced by uniformly random normalized vectors. The predictions produced with this model received a Kaggle score of 0.7242, so the Word2vec default model's word embeddings do indeed carry semantic information.

As to whether small perturbations like misspellings would greatly affect the results, the answer depends on the amount of words misspelled as each misspelled word would be ignored. So if a small amount of words are misspelled, that would not matter, but if all the words were misspelled that would be a problem.

## Justification

The benchmark model received a 0.8687 score from Kaggle for its predictions on the testing dataset while the cross-validated logistic regression model using feature vectors from the default Word2vec model received a 0.8204 score. Ie. the model performed 5.6% worse than the benchmark model's score. The conclusion of this project is thus that feature vectors created with the Word2vec word embedding model do not perform well on the Bag of Words Meets Bags of Popcorn Kaggle competition.

# V. Conclusion

## Free-Form Visualization

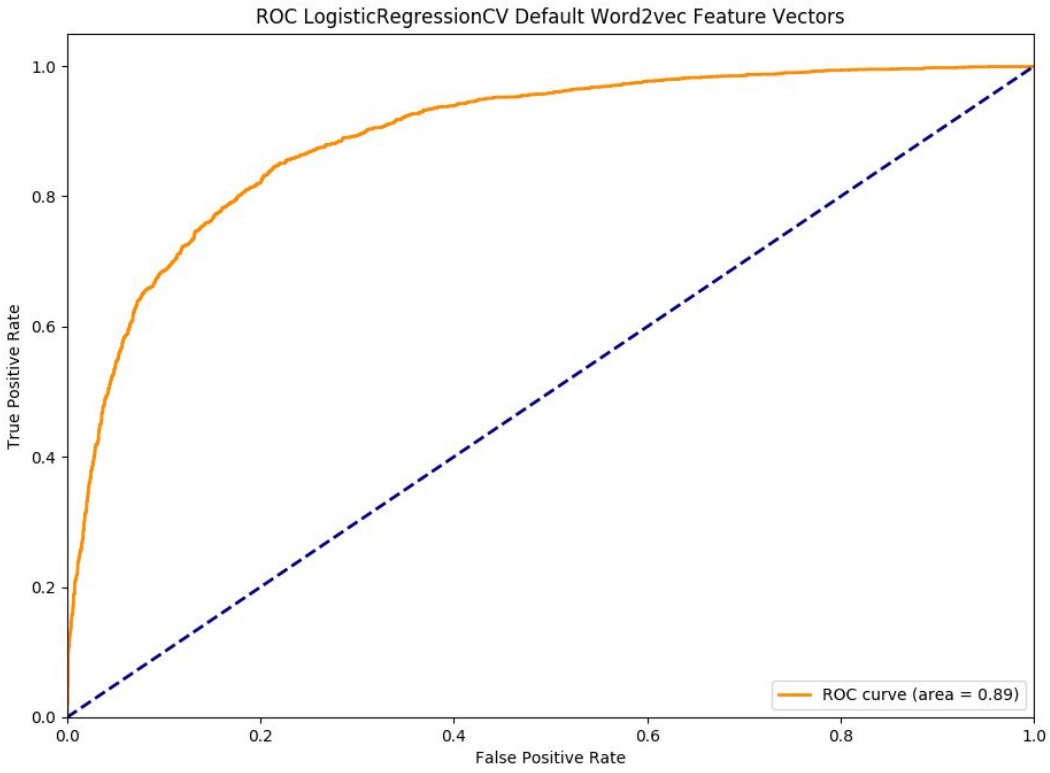


Fig. 3. ROC curve of the logistic regression model using mean feature vectors from the default Word2vec training.

Fig. 3. displays a plot of the receiver operating characteristic curve produced by evaluating the cross-validated logistic regression model that uses mean feature vectors from the default Word2vec training on the validation dataset. It shows that achieving a high true positive rate is only possible at the cost of a high false positive rate. So if an 0.5 false positive rate is tolerable, we can identify almost all of the positive reviews. Conversely, if an 0.2 true positive rate is tolerable, we can be almost certain that we will not classify any of the negative reviews as positive ones.

The following table shows the 9 closest word vectors to the words “good” and “bad” from the default Word2vec training:

| **good** | **bad** |
|---|---|
| decent | stupid |
| nice | terrible |
| great | *good* |
| fine | horrible |
| *bad* | keep |
| brief | georg |
| excellent | lame |
| ok | cruel |
| fantastic | ridiculous |

Both “good” and “bad” are within the nine closest synonyms of each other according to the Word2vec model, because both words are adjectives that show up mostly in the same position. This explains why the model has a high false positive rate.

## Reflection

The goal of the project was to examine whether feature vectors created with the Word2vec word embedding model perform well on the Bag of Words Meets Bags of Popcorn Kaggle competition. The solution consisted of training a Word2vec model on the unlabeled training data provided by Kaggle, creating feature vectors for reviews by taking the mean of all the word vectors in a review and then training a logistic regression model using these feature vectors to discriminate between positive and negative reviews. The conclusion is that creating averaged feature vectors from Word2vec embeddings trained on the unlabeled movie review data provided by Kaggle is not a good approach to solve the movie review sentiment analysis task.

An interesting aspect of the project was how well logistic regression performed on the task compared to a support vector classifier and a random forest classifier which are generally thought to be more sophisticated models. It is possible that the logistic regression’s good performance is related to the fact that the Word2vec model uses logistic regression during training to discriminate between words in the current context and other words.

The main challenge of the project stems from the subtleties of natural language: The meaning of words depend on their immediate and larger context. The word “good” has an opposite meaning if it is preceded by the word “not”, and even the presence of the phrase “not good” is not evidence that a review expresses a negative sentiment, as it might be part of a sentence that

explains the plot such as “The opening scene of the movie finds Jane in a restaurant explaining her friend how the movie she watched yesterday was not good at all.”

An additional difficulty in the project was that I have not found word embeddings that were trained on a larger corpus for the TensorFlow Word2vec model which could have increased the model’s accuracy.

## Improvement

As explained in the previous parts, the main weakness of the solution presented here is that it can not distinguish between words if they appear in the same positions and that it does not take their context into account when deciding on semantics. Both problems are addressed by (Miyato et al., 2016) that proposes adversarial training and a long short-term memory network achieving a 5.91% error rate on the same task. Another simple improvement could be using a larger corpus for training the word vectors. Furthermore, the best submissions to the task on Kaggle all use an ensemble of models and fine tune their weights on a validation dataset according to this thread, so it would be desirable to use an ensemble of methods to produce a competitive model.

# VI. References


Bishop, Christopher M. *Pattern Recognition and Machine Learning.* New York: Springer, 2006. Print.

Chapman, Angela. *Bag of Words Meets Bags of Popcorn | Kaggle.* N.p., 2014. Web. 03 June 2017.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). "Learning Word Vectors for Sentiment Analysis." *The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).* (link)

Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

Miyato, Takeru, Andrew M. Dai, and Ian Goodfellow. "Adversarial Training Methods for Semi-Supervised Text Classification." *arXiv preprint arXiv:1605.07725* (2016).

Nakov, Preslav, et al. "SemEval-2016 task 4: Sentiment analysis in Twitter." *Proceedings of SemEval* (2016): 1-18.

Rong, Xin. "word2vec Parameter Learning Explained." *arXiv preprint arXiv:1411.2738* (2014).

Wasserman, Larry. *All of Statistics: A Concise Course in Statistical Inference.* New York: Springer, 2004. Print.

Wikipedia contributors. "Receiver operating characteristic." *Wikipedia, The Free Encyclopedia.* Wikipedia, The Free Encyclopedia, 25 May. 2017. Web. 1 Jun. 2017.

Wikipedia contributors. "Word embedding." *Wikipedia, The Free Encyclopedia.* Wikipedia, The Free Encyclopedia, 24 Apr. 2017. Web. 27 May. 2017.