A Survey of Word2Vec Inversion Methods in Topic Recognition Tasks

Sarah Wiegreffe
Department of Computer Science
College of Charleston
66 George St
Charleston, SC 29424
wiegreffesa@g.cofc.edu

Paul Anderson, PhD
Department of Computer Science
College of Charleston
66 George St
Charleston, SC 29424
andersonpe2@cofc.edu

Abstract

An extension of Google's word vector embedding algorithm, Word2Vec (Mikolov et al., 2013), Word2Vec Inversion (Taddy, 2015a) is a simple and robust technique that allows for classification tasks such as topic recognition of variable-length inputs including sentences and documents. The method uses a simple averaging of sentence scores to classify the document, and demonstrates that performance is similar to other traditional methods, while better capturing word similarity and importance. However, Word2Vec Inversion is sensitive to bias from strongly misclassified sentences when factored into the average. In this paper an overview of various alternative methods for combining sentence scores is presented, and their performance compared to Word2Vec Inversion is reported, in an effort to better capture overall review classification in a way less sensitive to sentence bias.

1 Introduction

Topic recognition of a text, such as determining the sentiment of a movie review, is a popular topic of study in the field of natural language processing, in many ways because of the dissimilarity between the quirks of language and communication that express sentiment and the capabilities of computational machines. Traditional machine learning algorithms such as Random Forest require input data to have a fixed-length number of features, but words and documents have variable numbers of characters and words. Various methods such as Bag of Words have been developed to transform variable-length documents into fixed-length objects and subsequently use them in a classification task. Issues with these methods include a disregard for word order, similarity, and importance (treating every word in a document with the same weight or documents containing the same words in different orders having the same representation, for example).

Word embedding techniques such as Google's Word2Vec algorithm proposed by Mikolov et al. (2013) map each word in a high-dimensional vector space to account for similarity, generating fixed-length distributed representations that can be passed on as input to a machine learning algorithm. Taddy extends this technique by proposing a method called Word2Vec Inversion (2015a), which trains two topic-specific Word2Vec models and uses a Bayesian likelihood score a new document receives from each model as the means by which it is classified, skipping the preprocessing step altogether.

Taddy's implementation uses the Python gensim Word2Vec package (2015b), and works by building a vocabulary and subsequently training a basemodel on all sentences in the training set, then training two separate, class-specific copies of the basemodel using only the training data of that class (in this case, positive or negative sentiment) from a dataset of Yelp movie reviews. Each of the two models scores a new (test) document with the likelihood of it belonging to the model,

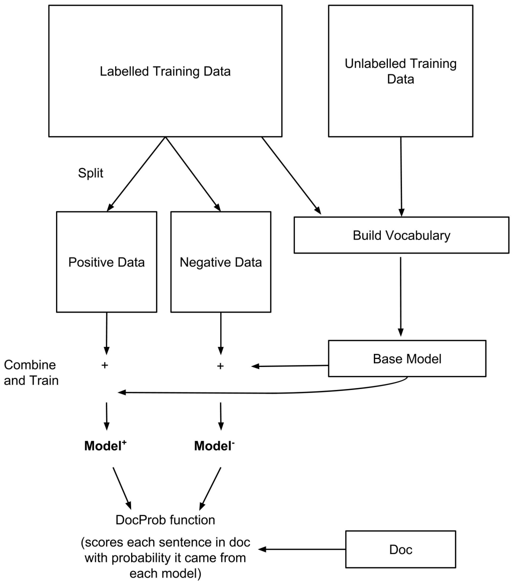

Figure 1: A high-level overview of the Word2Vec Inversion algorithm.

and the document can thus be classified as belonging to the class for which it receives the highest likelihood score. However, the Word2Vec Inversion algorithm scores sentences rather than whole documents, so the two models actually provide a likelihood score for each sentence in the review, which are averaged to obtain a document likelihood (see figure 1).

2 Motivation

The motivation for this work comes from an analysis of movie reviews from the Kaggle IMDb dataset (2014) (and parsed using the Kaggle Word2Vec Utility (Kan, 2014)) misclassified by Word2Vec Inversion in a basic sentiment analysis task. It was determined that the most common characteristic shared by misclassified reviews (both positive and negative) is having one or a few misclassified sentences within a review that received very strong likelihood scores of being of the opposite sentiment of the review as a whole, skewing the averaged likelihood. It was also observed that sentences of an opposite sentiment of the review as a whole were often located at the beginning or end of the review. For example, a movie reviewer might critique certain aspects of a film at the beginning of his or her review, but conclude with a feeling of ambivalent positivity towards it. A strong negative first sentence such as this was observed to have caused misclassification of the review in some instances.

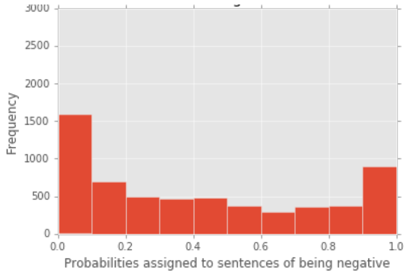

Figure 2: Probability Distribution of Positive Sentence Scores among Misclassified True Positive Reviews.

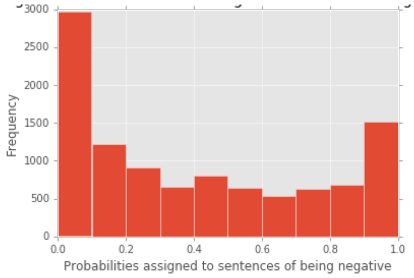

Figure 3: Probability Distribution of Negative Sentence Scores among Misclassified True Negative Reviews.

Probability distributions were generated to visualize this trend (see figures 2 and 3). By looking at the sentence score distribution among all misclassified reviews, useful information comes to light. For example, among misclassified positive but true negative reviews, almost two times more sentences were assigned a 0 to 0.1 likelihood of being negative (and thus a 0.9 to 1.0 likelihood of being positive) than a 0.9 to 1.0 likelihood.

3 Methods

3.1 Exploration

Initial alternative methods to sentence averaging developed in an effort to reduce sentence bias include using the maximum and median sentence scores to classify. (Because the likelihoods assigned by the positive sentiment model and the negative are normalized to sum to one in Taddy's implementation (2015b), selecting the maximum score, minimum score, maximum difference between positive and negative likelihood, or minimum difference, are all equal and effectively result in the same document score).

Initial Results

Accuracy scores from using the maximum sentence score (MaxProb) and median methods on the Kaggle IMDb movie reviews with a simple training/testing split of 50/50 and 5-fold cross validation are found in Table 1. Performance improvements were not seen, rather performance remained about the same regardless of technique.

After once again analyzing misclassified reviews from the MaxProb method and the baseline averaging, it became evident that many reviews had sentences scoring very highly (with greater than 0.9 likelihood) of belonging to both classes. Thus, selecting the maximum of the two was not valuable in determining the true relationship of that high scoring sentence to the remainder of the review and its overall sentiment.

3.2 Development of the Random Forest of Distribution Features Technique

Based on how similar the results obtained from the exploratory study were, we set out to answer the hypothesis of whether employing a machine learning algorithm to find the ideal means of combining sentence scores could improve prediction accuracy. Using machine learning to discern the most important features of the reviews or positions of the sentences instead of attempting to do so manually allows for generalization of the technique to other topic recognition tasks.

From the output of the Word2Vec Inversion model, each sentence received a probability score from the negative model. Various summary scores were generated from groups of these sentence scores on an individual review level, including the mean, minimum, and maximum scores, the standard deviation of scores, scores of the first and last sentences, number of sentences in the review, sigmoidal approximation functions with k=10 and k=20, and positions of the minimum and maximum scores in the review. These generated distribution features were compiled as a fixed-length feature vector of size 10 for each review. The feature vectors were then used to train decision tree ('DT') and random forest ('RF') learners (see figure 4).

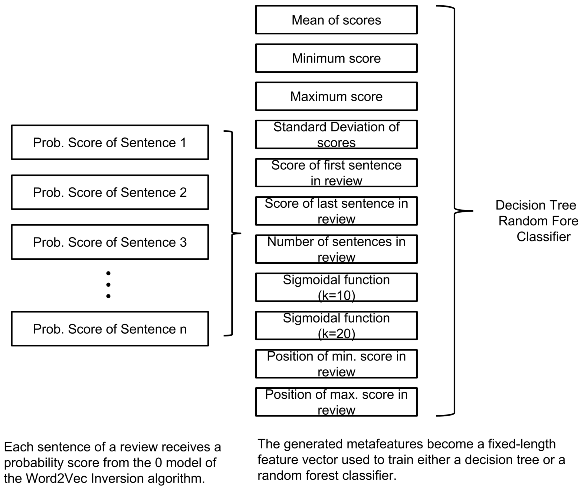
Prob. Score of Sentence 1
Prob. Score of Sentence 2
Prob. Score of Sentence 3
Prob. Score of Sentence n
Mean of scores
Minimum score
Maximum score
Standard Deviation of scores
Score of first sentence in review
Score of last sentence in review
Number of sentences in review
Sigmoidal function (k=10)
Sigmoidal function (k=20)
Position of min. score in review
Position of max. score in review
Decision Tree or Random Forest Classifier
Each sentence of a review receives a probability score from the 0 model of the Word2Vec Inversion algorithm.
The generated metafeatures become a fixed-length feature vector used to train either a decision tree or a random forest classifier.

Figure 4: An overview of the Distribution Features technique.

The methods were tested on three datasets: IMDb movie reviews (imd, 2014), fine food reviews from Amazon (McCauley and Leskovec, 2013), and tweets taken from Twitter (Naji, 2012). The Kaggle IMDb movie reviews had a size of 100,000 records, class labels of either positive or negative sentiment, and were multiple sentences long on average. The Twitter sentiment analysis dataset was randomly subsetted down to 99,999 records for computational feasibility and had the same class labels, but the records were shorter in length- many only parsed to one sentence. The Amazon fine food reviews dataset (McCauley and Leskovec, 2013) was multi-class, with labels of one to five star ratings assigned by the writers of the reviews, and, following a subsetting of the five-star class to make the difference in class size more equal, had 255,332 records.

Technique	Mean (Baseline)	MaxProb	Median
Score	0.8738	0.8551	0.8540

Table 1: Exploratory Method Results for Combining Sentence Scores for Kaggle's IMDb Movie Reviews dataset.

4 Results

Tables 2 through 6 illustrate various performance measures of the methods on the three aforementioned datasets. The results are presented as 95% confidence intervals generated from 25 runs of a 5 fold cross-validation loop for each model. For the multi-class Amazon fine foods dataset, scores were calculated using a one-versus-all approach for each class, and presented in the table as the average performance across all five class models. No Word2Vec Inversion with a decision tree or random forest classifier scores are reported for this dataset, due to the 5-class scoring rendering the task too computationally expensive.

For the Kaggle IMDb movie reviews dataset the improvement in accuracy achieved from using the Random Forest with Distribution Features is roughly 0.5% over Word2Vec Inversion, but this improvement is lost when comparing other performance measures. For the Twitter dataset, the Decision Tree outperforms the Random Forest method, and outperforms baseline Word2Vec Inversion by the slightest amount when considering accuracy, but allows for significant improvements in precision, recall, and F1-scores. However, the Decision Tree and Random Forest methods are less suitable for the Twitter dataset because so many of the reviews are only one sentence, which does not allow for meaningful features about the distribution.

We also achieve relatively good classification accuracy on the Amazon fine foods dataset using Word2Vec Inversion, especially considering that 90% performance is considered ideal in sentiment analysis tasks due to about 10% of the data being generally accepted as being disagreed upon by humans themselves (Naji, 2012).

5 Conclusion

Preliminary results support that the Decision Tree and Random Forest with Distribution Features techniques, which stemmed from a goal of finding a more optimal means of combining sentence score output from Word2Vec Inversion into a classification, outperform baseline Word2Vec Inversion, but are not decisive. However, the results support that Word2Vec Inversion should not be discredited as a well-performing algorithm for classification. Completing other topic recognition tasks apart from sentiment analysis on datasets with a more diverse range of document lengths and vocabularies, such as determining the journal of publication of medical abstracts, will provide further insight into how these results are significant in real-world applications, as would further expanding processing power to be able to calculate on larger datasets and those with many classes.

6 Note

All code used in implementing this work can be found at `https://github.com/sarahwie/paper_final_code`.

References

[imd2014] 2014. Bag of words meets bags of popcorn [data files]. Available from `https://www.kaggle.com/c/word2vec-nlp-tutorial/data`.

[Kan2014] Wendy Kan. 2014. Kaggleword2vecutility.py. Available from `https://github.com/wendykan/DeepLearningMovies/`.

[McCauley and Leskovec2013] Julian McCauley and Jure Leskovec. 2013. From amateurs to connoisseurs: Modeling the evolution of user expertise through online reviews. WWW. Datasets available from `http://snap.stanford.edu/data/web-FineFoods.html`.

Dataset	**Word2Vec Inversion**	**Inversion + DT**	**Inversion + RF**
Movie reviews	(0.8763, 0.8775)	(0.8791, 0.8804)	(0.8801, 0.8814)
Fine Food reviews	(0.8367, 0.8371)		
Twitter	(0.7321, 0.7331)	(0.7367, 0.7376)	(0.6982, 0.6993)

Table 2: Accuracies from 25 times 5-fold cross validation on the datasets.

Dataset	**Word2Vec Inversion**	**DT + dist features**	**RF + Dist features**
Movie reviews	(0.8763, 0.8775)	(0.8791, 0.8804)	(0.8801, 0.8814)
Fine Food reviews	(0.7309, 0.7315)		
Twitter	(0.7321, 0.7331)	(0.7367, 0.7376)	(0.6982, 0.6992)

Table 3: Areas under the ROC Operating Curve.

Dataset	**Word2Vec Inversion**	**Inversion + DT**	**Inversion + RF**
Movie reviews	(0.8805, 0.8815)	(0.8808, 0.8819)	(0.8804, 0.8817)
Fine Food reviews	(0.7196, 0.7213)		
Twitter	(0.5708, 0.5714)	(0.7348, 0.7361)	(0.6914, 0.6923)

Table 4: F1-scores.

Dataset	**Word2Vec Inversion**	**Inversion + DT**	**Inversion + RF**
Movie reviews	(0.8516, 0.8549)	(0.8681, 0.8721)	(0.8782, 0.8798)
Fine Food reviews	(0.7538, 0.7548)		
Twitter	(0.5740, 0.5751)	(0.7389, 0.7408)	(0.7069, 0.7083)

Table 5: Precisions.

Dataset	**Word2Vec Inversion**	**Inversion + DT**	**Inversion + RF**
Movie reviews	(0.9090, 0.9127)	(0.8910, 0.8951)	(0.8822, 0.8841)
Fine Food reviews	(0.6879, 0.6914)		
Twitter	(0.5689, 0.5708)	(0.7295, 0.7330)	(0.6762, 0.6774)

Table 6: Recalls.

[Mikolov et al.2013] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimations of word representations in vector space. ICLR.

[Naji2012] Ibrahim Naji. 2012. Twitter sentiment analysis training corpus (dataset).

[Taddy2015a] Matt Taddy. 2015a. Document classification by inversion of distributed language representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 45–49, Beijing, China, July. Association for Computational Linguistics.

[Taddy2015b] Matt Taddy. 2015b. w2v-inversion.ipynb. Available from `https://github.com/TaddyLab/deepir/`.