Integrating Rich Document Representations for Text Classification

Suqi Jiang, Jason Lewris, Michael Voltmer, Hongning Wang
University of Virginia, sj4wp, jl4pt, mv5vf, hw5x@virginia.edu

Abstract - **This paper involves deriving high quality information from unstructured text data through the integration of rich document representations to improve machine learning text classification problems. Previous research has applied Neural Network Language Models (NNLMs) to document classification performance, and word vector representations have been used to measure semantics among text. Never have they been combined together and shown to have improved text classification performance. Our belief is that the inference and clustering abilities of word vectors coupled with the power of a neural network can create more accurate classification predictions. The first phase our work focused on word vector representations for classification purposes. This approach included analyzing two distinct text sources with pre-marked binary outcomes for classification, creating a benchmark metric, and comparing against word vector representations within the feature space as a classifier. The results showed promise, obtaining an area under the curve of 0.95 utilizing word vectors, relative to the benchmark case of 0.93. The second phase of the project focused on utilizing an extension of the neural network model used in phase one to represent a document in its entirety as opposed to being represented word by word. Preliminary results indicated a slight improvement over the baseline model of approximately 2-3 percent.**

Index Terms - Natural Language Processing, Text Classification, Text Mining, Word2vec

INTRODUCTION

The ability to sift through massive amounts of unstructured text data in a meaningful and impactful way can yield tremendous value towards businesses across a multitude of domains. One field which derives value from unstructured text data is text mining. Text mining is concerned with yielding quality information from unstructured text, processing it in a way that can be consumed by computers and statistical models, with the goal of identifying patterns and knowledge to drive value [1]. This high quality information can then in turn be used for a variety of problems, including machine learning based classification. This paper explores state of the art mechanisms for capturing information from unstructured text for the purposes of classification via word vectors.

Before exploring details of the mechanisms used to develop classification models, a broader use case context needs to be addressed. This paper will focus on developing models for categorization by capturing sentiment polarities in text data. This same process for analyzing user sentiment can be applied towards business operations where having a deeper understanding of one's customers can generate value. One such scenario where this can be useful is in the prediction of whether a customer is at risk for departing from an organizations products or services by analyzing the customer text in transactions with the company. Beyond customers, this same methodology can be used for organizations internally to assess the likelihood of any given employee departing the organization [2]. Firms have a wealth of text data related to their employees through tools such as Slack, SharePoint and corporate e-mail, all of which are potential candidates for applying the same principles outlined in this paper towards a learning objective most valuable to any given firm.

One mechanism used to capture the complexities within textual data is a neural network method called word2vec [3]. The goal of word2vec is to predict surrounding words, given a word. The weights of the neural network are calculated using back propagation and stochastic gradient descent [3]. The result of utilizing and training a word2vec model is a corresponding vector representation of words, with similar words with similar meanings having similar vector representations [3]. Using these word2vec vector representations of words as inputs into a classification model is expected to yield superior results over simpler methods, such as bag-of-words representation, due to word2vec's superior inference and clustering capabilities. Furthermore, an unsupervised word2vec model for supervised classification tasks assists with overcoming scalability issues while retaining the complexity that naturally occurs within written language, since the vector representations of words can be manipulated and compressed, highlighting signals within the data while suppressing noise. This enables users to streamline their workflows, reducing the time required to obtain substantial results.

Various document representations will be tested against benchmark cases. Classification rates from a bag of words classification model will be used as the benchmark case to test the effectiveness of word2vec vectors within the feature space.

Subsequent sections of this paper will explore this methodology in greater detail including reviewing previous

978-1-5090-0970-1/16/$31.00 ©2016 IEEE

research within this problem set. A detailed outline of the process used within this engagement will be displayed, as well as in depth coverage of the results obtained. The results obtained within this paper can be used across a multitude of knowledge domains covering text.

LITERATURE REVIEW

word2vec introduced many different ways of extracting meaningful vectors from words. Two architectures are the Continuous Bag of Words (C-BOW) method and Skip-Gram method [3]. The two main differences between these methods are that the C-BOW predicts a word given its surrounding words, and the Skip-Gram predicts surrounding words given a single word. Research has shown that the Skip-Gram continually outperforms the C-BOW structure [3]. Additionally, Mikolov et al suggested that the Skip-Gram coupled with a negative sampling optimization technique performs particularly well in a distributed setting [4]. This was tested on a dataset of Google News that contains about 6 billion tokens against several different models. In this test where accuracy was used as the primary metric of evaluation, the Skip-Gram method performed best with a score of 53.3% [3]. This outperformed several other neural network based models that have been used in the past for Natural Language Processing, the highest performing algorithm receiving an accuracy score of 24.6%. Not only has the Skip-Gram performed well on single machines, but in distributed architectures it out-performed other neural network models as well as more traditional methods like Latent Semantic Analysis [3].

There have been different alterations of word2vec that have been created and achieved similar results. One of which is GloVe. GloVe is a count-based model that learns its word vectors using dimensionality reduction via matrix factorization, instead of a neural network like word2vec. In certain cases, like on a Named Entity Recognition (NER) dataset with 1.6 billion unique words, and a specified dimensionality of 300, GloVe outperforms the skip-gram implementation of word2vec using accuracy of semantics as the evaluation metric [5]. Word2vec still tends to be the industry standard, despite these results.

Specifically for document classification tasks, the algorithm doc2vec was created as an alternative to word2vec. Mikolov and Le introduce the idea of "paragraph vectors" as a way to represent a larger body of text, rather than a single word. They use "paragraph" to refer to a sentence, paragraph, or whole document of text, as long as everything is specifically labeled. Doc2vec is structured very similarly as word2vec except that it has an additional parameter in the input layer that represents the paragraph where a given word is located in [6]. Compared to bag-of-words, and other neural network-based classifiers, doc2vec and its paragraph vectors result in very promising error rates.

The idea of clustering words for document classification has been done in the past using Latent Semantic Indexing [7]. In doing so, the dimensionality is reduced drastically while only incurring a minimal loss in accuracy.

METHODS

In order to compare any improvements word2vec features had on classification tasks, a baseline model was developed utilizing a bag of words representation of text as features in a classification model. This was done via a random sample of approximately 100,000 review documents from TripAdvisor and, and 50,000 review documents from a Yelp. Typically, a user would rate a hotel on TripAdvisor or a restaurant on Yelp on a 1-5 "star" scale. This target output indicates the level of satisfaction an individual had towards any given hotel or restaurant, with a 5 indicating completely satisfied and a 1 indicating completely dissatisfied. This being the target output, and due to severe class imbalances, it was changed to a binary classification task of 0 indicating a poor review, and a 1 indicating a positive review. The following alignment of target outputs was conducted:

TABLE I
ADJUSTED RESPONSE VALUES

Original Output	Adjusted Output
1	0
2	0
3	0
4	1
5	1

Given the above adjustments, it is also important to highlight some of the differences among the two datasets under analysis. Within the TripAdvisor review dataset, the average length of a given document, or review, was 86 words. In the Yelp dataset, the average length of a review was 67 words. Additionally, the distribution of the target variable for TripAdvisor was approximately 73% for the target output of positive. For the Yelp dataset, 77% of the target output consisted of positive. This information is summarized in Table II.

TABLE II
TRIPADVISOR VS YELP

TripAdvisor	Yelp
86 words per review	67 words per review
73% Target Value "1"	77% Target Value "1"
27% Target Value "0"	23% Target Value "0"

With the baseline model acquired, the focus shifted towards utilizing more robust methods of representing

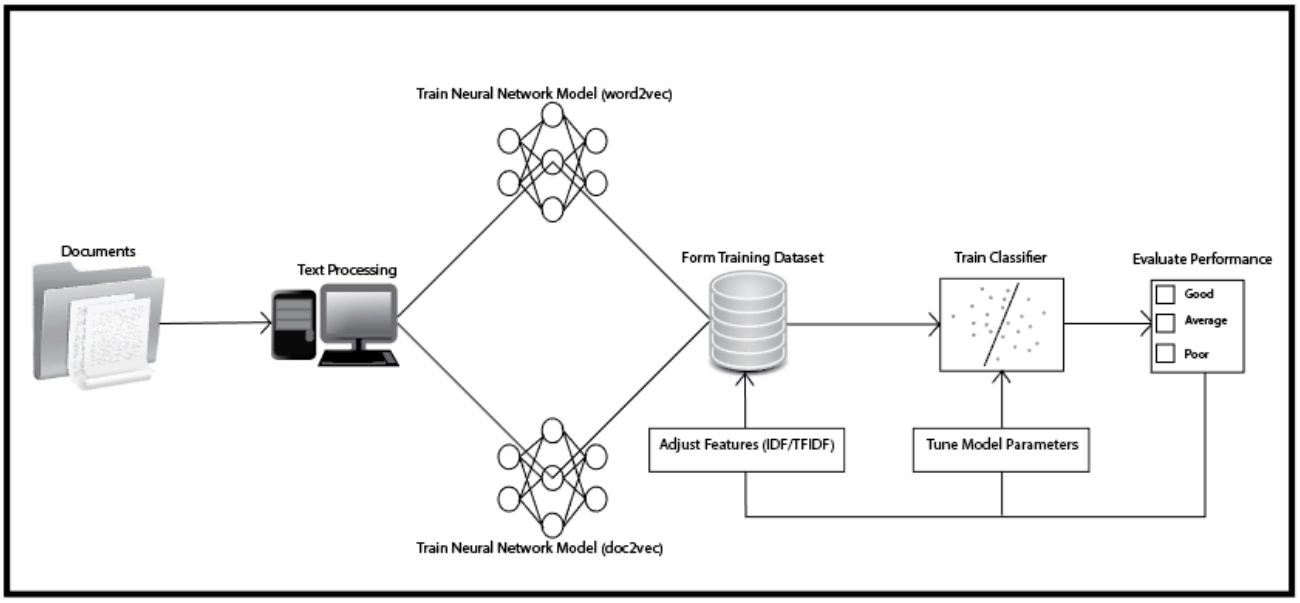

FIGURE I
PROCESS FRAMEWORK

documents for classification tasks. A word2vec model was trained over the entire TripAdvisor and Yelp datasets using the Gensim package in Python. The parameters utilized for training consisted of incorporating the skip-gram variant of the algorithm, a window size of 10, and down sampling set to 10^{-3}. This word2vec model was then utilized to generate individual document vectors based on each word within the review. Each word within a review consisted in a vector of length 100, the output from the word2vec model when fed an individual word. These word vectors were averaged to create a vector that represents the sentiment of the review.

Similar to this process, we applied the doc2vec algorithm in the Gensim package to these two datasets as well. This will provide a good comparison to the aforementioned averaged word2vec results. Doc2vec is an extension of word2vec where the model creates document vectors instead of individual word vectors. A doc2vec model was trained across the random sample of 100,000 reviews, and each document vector was utilized as input features within the dataset.

In an attempt to better capture important words within the data, a weighting schematic beyond simple average was utilized for any given word. The weighting schematic initially attempted was a relative weight according to the inverse document frequency (IDF) of a given word. The IDF is a way to assess the commonality or rarity of a word across the entire corpus. It is scaled by taking the log of the result of dividing the total number of documents in the corpus by the number of documents where the specific word appears. We use an inverse document frequency equation, a weighting schematic, where N represents the number of documents in the collection, and t is the number of documents containing term t [10], in our work.

To further enhance the predictive capabilities, it was found that appending a bag-of-words as features to the word2vec document reviews was worthwhile. Each word within the bag-of-words was weighted by its TF-IDF score. This calculation is similar to the IDF mentioned previously, although each word is multiplied by the number of times it occurs in the document. Theoretically, this should capture word "importance" based on the hypothesis that the most important words occur most often.

Every test conducted was evaluated using area under the curve (AUC) with ten-fold cross-validation, using logistic regression as the classification technique. A visual summary of the methodology undertaken throughout this engagement can be seen in Figure I.

The driving force behind the previously mentioned ways of representing documents for text classification was to do a thorough exploration in an attempt to consolidate and compare different methods with the goal of classification. This can enable others to view results, and extend sections based on their own work and domains. Prior to this engagement, there was little done in the way of a consolidated place with various methods for representing documents in this way.

RESULTS

The results of analysis on both the TripAdvisor and Yelp review datasets can be found below. It is important to note in all instances a logistic regression classification model was used. Furthermore, in all instances 10-fold cross validation and AUC metrics were used for comparative purposes.

The first classification model was built utilizing a bag of words representation of each review in the corpus of the TripAdvisor dataset. This resulted in an AUC metric of 0.93 obtained with a logistic regression classifier. The results of this can be visualized in Figure II below.

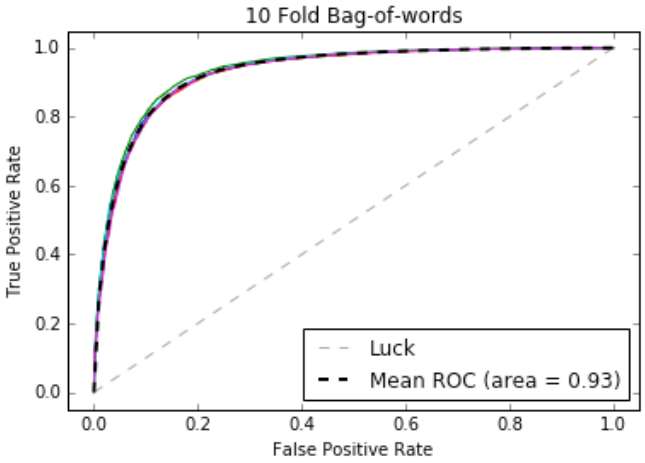

FIGURE II
BAG OF WORDS - TRIPADVISOR

Using a simple average of the corresponding word vectors within a logistic regression classification model, the results yielded a model with an AUC of 0.91, similar to the results found when using the bag of words model.

The IDF weighted model resulted in an AUC of 0.93, results of which are in alignment with previous models.

To further enhance the predictive capabilities of the logistic regression model, a new weighting mechanism, term-frequency, inverse document frequency, or TF-IDF, was utilized as features being appended to the preexisting word2vec word vectors. This yielded an AUC of 0.95 using 10-fold cross validation, the results of which can be viewed in Figure III. This produced a slight improvement over the baseline model.

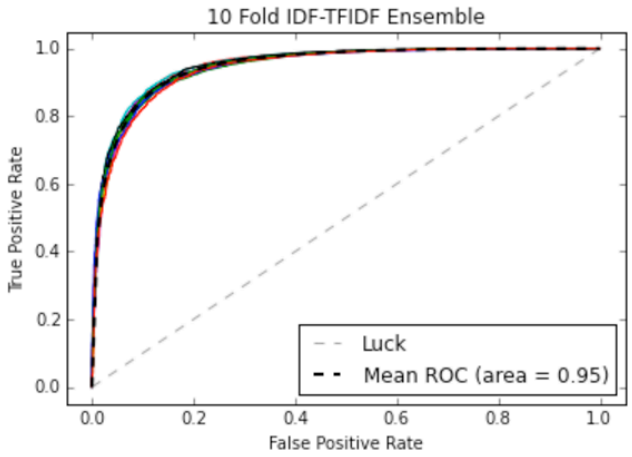

FIGURE III
WORD2VEC TF-IDF - TRIPADVISOR

Lastly, a different representation of reviews was utilized as features for classification purposes. The doc2vec extension of word2vec was used to represent reviews within the dataset. Yet again, a logistic regression model was utilized in order to assess the predictive capabilities of this new representation of documents as features. Figure IV shows the results of this with 10-fold cross validation. The results yielded an average AUC of 0.93.

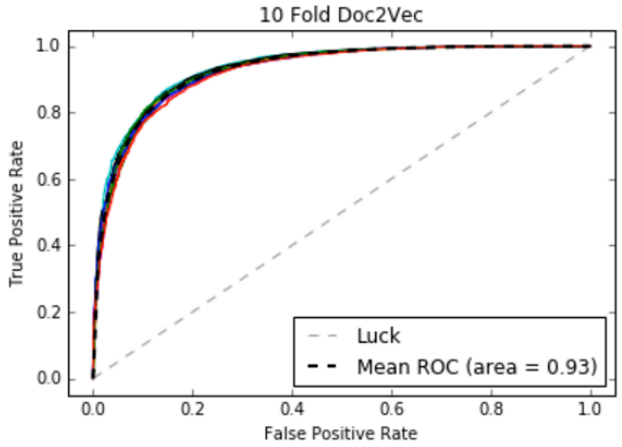

FIGURE IV
DOC2VEC - TRIPADVISOR

To validate the adequacy of word2vec and doc2vec representations of documents as features for classification purposes, the secondary Yelp dataset was analyzed in the same way outlined previously. To begin the analysis, a bag of words representation of documents was created utilizing a logistic classification model on the Yelp data. This baseline model yielded an AUC of 0.87, the results of which can be viewed in Figure V.

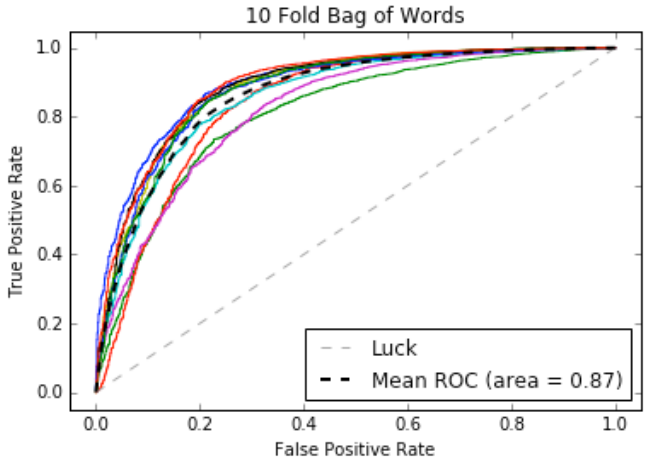

FIGURE V
BAG OF WORDS - YELP

Having the Yelp dataset baseline intact, a word2vec model was trained across the entire dataset. Using a similar mechanism for capturing word vectors for any given review, the feature space was generated using word2vec review vectors in conjunction with IDF, TF-IDF features. This expanded feature space was used for training and evaluating a model, yielding an AUC of 0.9, which can be visualized in Figure VI.

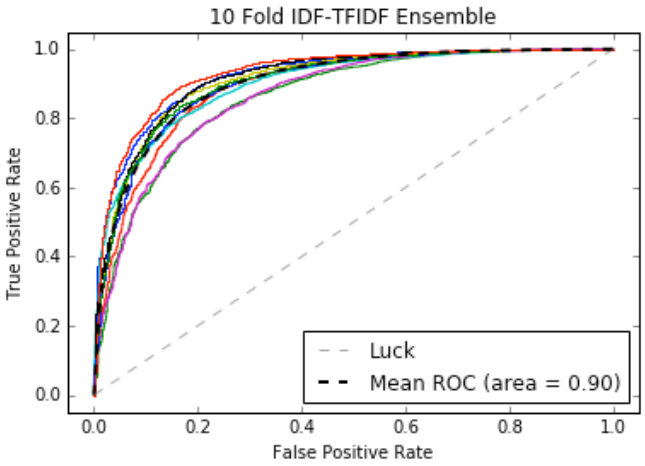

FIGURE VI
WORD2VEC TF-IDF - YELP

Reviewing the above, we can see an improvement over the baseline model for the Yelp dataset, using an enhanced representation of documents or reviews within the feature space.

Subsequent analysis on the Yelp dataset included the use of doc2vec. The results obtained with this method was an AUC of 0.92, an improvement over the baseline bag of words model, and a further improvement over the weighting mechanisms used with word2vec output.

In order to summarize the results obtained through the various methods of representing documents in the feature space for a classification model, Table III below highlights the results and methodologies.

TABLE III
RESULTS

Dataset	Methodology	AUC
TripAdvisor	Bag of Words	0.93
TripAdvisor	Word2vec Average	0.93
TripAdvisor	Word2vec Weighted IDF	0.93
TripAdvisor	Doc2vec	0.93
TripAdvisor	Word2vec TF-IDF	0.95
Yelp	Bag of Words	0.87
Yelp	Word2vec Average	0.86
Yelp	Word2vec Weighted IDF	0.86
Yelp	Doc2vec	0.92
Yelp	Word2vec TF-IDF	0.90

An interesting point to note is the discrepancy between the classification results obtained with the TripAdvisor dataset and the Yelp dataset. One possibility for this could be the difference in the average document length between the two datasets. TripAdvisor was approximately 20 words longer in length, enabling word2vec more training examples to capture the context of a given corpus.

CONCLUSION

The original hypothesis posed was word2vec and doc2vec representations of documents could yield additional predictive power for text classification tasks relative to a more traditional representation of documents such as bag of words. The hope was more advanced representations would enable learners to have a deeper understanding of any given word within a document and the relationship of one document relative to another. This deeper understanding can be used for a broad range of problem sets, such as analyzing customer text as they interact with organizations, in an attempt to classify customers at risk for departing the organizations services. Furthermore, this same approach can be applied towards internal business communication in an effort to gauge employee satisfaction and determine employees at risk for departing the organization.

In the analysis for this particular engagement, it was determined based on two distinct datasets containing user generated reviews for hotels and restaurants, that this type of document representation does not significantly improve classification results, with an exception for the TF-IDF weighting schematic of word vectors for TripAdvisor and Yelp, where a marginal improvement was yielded from the baseline model. However, it should be noted that slight improvements such as this can be relevant depending upon the domain of interest, where in some contexts slight improvements can be substantial. It is also important to note that utilizing word vectors for classification purposes through word2vec does degrade the interpretability of the model results.

In short, in most instances utilizing word2vec vector representations of words did not yield improvements for general sentiment classification tasks. When more advanced weighting schematics were utilized in conjunction with the word vectors, a slight improvement was generated. One must weigh the additional complexity in model interpretability with the gain in predictive power when deciding on how to represent documents in the feature space for classification tasks.

ACKNOWLEDGMENT

We would like to dedicate this space towards acknowledging our sponsor in this engagement, Capital One, who provided the resources and expertise to assist with this project. We would especially like to acknowledge and thank our Capital One liaisons, Adekunle Adediji, Will Franklin, and Chris Peterson. Lastly, we would like to acknowledge the University of Virginia Data Science Institute for their involvement with coordinating and facilitating this project.

REFERENCES

[1] Mooney, Raymond J., and Nahm, Un Y. 2003. "Text Mining with Information Extraction", *Multilingualism and Electronic Language Management*.

[2] Schrage, Michael. 2006. "Sentiment Analysis Can Do More than Prevent Fraud and Turnover." https://hbr.org/2016/01/sentiment-

analysis-can-do-more-than-prevent-fraud-and-turnover. Accessed: April 1, 2016.

[3] Mikolov, Tomas, Chen, Kai and Carrado, Greg et al. 2003. "Efficient Estimation of Word Representations in Vector Space." *arXiv preprint arXiv:1301.3781*.

[4] Mikolov, Tomas, Sutskever, Ilya, and Chen, Kai et al. 2013. "Distributed Representations of Words and Phrases and their Compositionality." *Advances in Neural Information Processing Systems*, pp. 3111 - 3119.

[5] Pennington, Jeffrey. Socher, Richard and Manning, Christopher D. 2014. "GloVe: Global Vectors for Word Representation." *EMNLP*, pp. 1532 - 1543.

[6] Le, Quoc and Mikolov, Tomas. 2014. "Distributed Representations of Sentences and Documents." *arXiv preprint arXiv:1405.4053.*

[7] Baker, L. Douglas and McCallum, Andrew K. 1998. "Distributional Clustering of Words for Text Classification." *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.*

[8] Ng, Andrew, Jordan, Michael I., and Blei, David. 2013. "Latent Dirichlet Allocation", *Journal of Machine Learning Research*, pp. 993 - 1022.

[9] Teh, Yee Whye, Jordan, Michael I., and Beal, Michael J. et al. 2006. "Hierarchical Dirichlet Processes"' *Journal of the American Statistical Association*.

[10] Manning, Christopher, Raghavan, Prabhakar, and Schutze, Hinrich. 2008. "Inverse Document Frequency." *Introduction to Information Retrieval,* New York: Cambridge University Press.

AUTHOR INFORMATION

Suqi Jiang, Student, Data Science Institute, University of Virginia.

Jason Lewris, Student, Data Science Institute, University of Virginia.

Michael Voltmer, Student, Data Science Institute, University of Virginia.

Hongning Wang, Ph.D, Assistant Professor of Computer Science, University of Virginia.