Efficient Word2Vec Vectors for Sentiment Analysis to Improve Commercial Movie Success

Yash Parikh(✉), Abhinivesh Palusa, Shravankumar Kasthuri, Rupa Mehta, and Dipti Rana

Sardar Vallabhbhai National Institute of Technology, SVNIT, Surat 395007, India
{yashparikhnitsurat, pabhinivesh, kasthurishravankumar}@gmail.com, {rgm, dpr}@coed.svnit.ac.in

1 Introduction

Movie industry has turned huge today, with a lot of money put at stake by the producer. Along with this, marketing strategies can be planned and improved dynamically according to the sentiments of the users available through their reviews on prerelease events like trailer, music launch, and other promotions. Better marketing can guarantee the producer at least a good opening whatever the story might be. This is where sentiment analysis [1] becomes useful. Analyzing the sentiments of the reviews has been worked upon since long, and the algorithms for sentiment polarity classification used include tf-idf [2], word2vec [3], and doc2vec [4].

Doc2vec provides pretty high accuracy each time, considering the area of sentiment polarity classification. But there are two limitations to it, namely, high space complexity to store paragraph vectors and high running time. In this paper, both these limitations are overcome by using a modified approach built on top of word2vec algorithm, which improves the classification accuracy considerably as compared to word2vec and gives comparable and sometimes even better results than doc2vec [4].

In this paper, in the first phase, gross is predicted by taking the attributes that the producers have, just after finishing the shooting of the movie, which can help them to plan the marketing strategies initially. In the second phase, sentiment analysis of the prerelease reviews is done at regular intervals using our proposed modified approach on word2vec and is compared with other techniques, which will help producers to plan and change their marketing strategies later on and will guide the distributors as to which movies are worth investing, considering the interest of people.

The rest of the paper is organized as follows. In Sect. 2, previous and related works are discussed, which are employed in the model. In Sect. 3, the implementation of the gross prediction is discussed followed by the implementation of sentiment analysis on the user reviews of movies in Sect. 4. In the same section, details of our new approach to improve the classification accuracy of word2vec vectors are given. The conclusion and possible future work are mentioned in Sect. 5.

S. Bhattacharyya et al. (eds.), *Advanced Computational and Communication Paradigms*, Lecture Notes in Electrical Engineering 475,
https://doi.org/10.1007/978-981-10-8240-5_30

2 Related Work

While a lot of work has been done in movie prediction in terms of hit or flop [4, 5], what actually matters to the producer is the profit that he earns on the movie. The proposed model in this paper is created using various machine learning techniques for classification. First, the data obtained from Internet Movie Database (IMDB) [4] are cleaned using techniques of binning, global value replacement and are combined into one file [6]. The techniques then used for classification include decision trees [6], random forest classifier [6], gradient boosting [6], Gaussian Naïve Bayes [7], logistic regression [8], and linear Support Vector Machine (SVM) [7].

The second phase focuses on sentiment analysis of movie reviews. Formerly, the major focus was on the analysis of the reviews after the release of the movie, whereas we focus a very different practical application that helps a producer as well as a distributor to increase their profits [5]. While sentiment analysis is done at word-level, phrase-level, and document-level, our focus is on document-level sentiment analysis as we want to delineate the sentiment of one entire movie review at once. For that, initially we model the words into vectors by utilizing various vectorization techniques like term frequency-inverse document frequency (tf-idf) [2], word2vec [3], and the latest doc2vec [4] utilizing paragraph vectors. In this paper, the expected rating obtained by Potts [9] is integrated on top of word2vec into the final vectors, which is utilized as input to various classifiers used in the first phase along with neural networks [10] and stochastic gradient descent [7]. As the expected rating obtained by Potts is signed, i.e., is negative for negative sentiment words and vice versa, this approach will be beneficial to sentiment polarity classification problems.

3 Proposed Model for Gross Prediction

This section focuses on the preparation of the model before the release of the movie.

3.1 Data Collection and Preprocessing

The data are collected from the open source repositories by imdb.com. All the files are available as X.list files, which are preprocessed and converted in the required form. The data use the following attributes for model preparation (Fig. 1):

- Actors' list
- Actors' popularity
- Director's popularity
- Genre
- Budget
- Release year
- Gross

Fig. 1. "Gross prediction" model workflow

Gross attribute is utilized as a training element for the model. All the above data are converted in the form of a.csv file. Next, preprocessing of the data aggregated is done. The preprocessing workflow is shown in Fig. 2.

In addition to the workflow shown in Fig. 2, genres attribute is also preprocessed by splitting and considering an optimized set of genres, which gives better classification results without increasing dimensionality much. Now, the term profit is subjective for different producers. But, one thing that the distributors as well as producers have their eye on is the ratio of Gross attribute to that of Budget as everyone wishes to recover what they spent and earn profit in its multiples.

Thus, using the above logic, a new attribute "Norm_Gross" is created that would act as our target attribute to be predicted, i.e., the ratio of "Gross" attribute to that of "Budget" attribute. This would as a result be an attribute with continuous distribution. But it can be modified to be distributed in different classes to estimate commercial success at the very beginning.

(1) Considering four classes

- Class 0: Range [0, 0.5)
- Class 1: Range [0.5, 1)
- Class 2: Range [1, 5)
- Class 3: Range [5, ∞)

(2) Considering three classes

- Class 0: Range [0, 1)
- Class 1: Range [1, 5)
- Class 2: Range [5, ∞)

Outliers tend to distort the model by overfitting. If the value of Norm_Gross is greater than 5, this value corresponds to an extraordinary or overwhelming success, which cannot be accounted for previously. Hence, these are outliers and are to be removed before applying modeling to the dataset. Thus, class 3 can be omitted from way number 1 of classifying and class 2 from way number 2 of classification. But just to ensure that the removal of outliers yields better results, the accuracy of a model is tested on it.

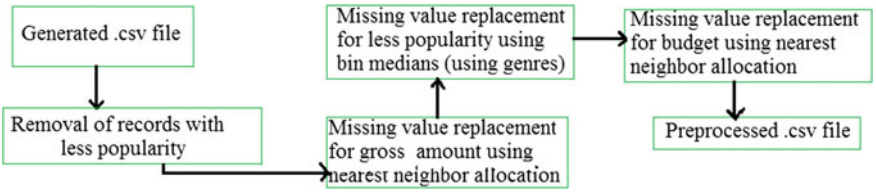

Fig. 2. Preprocessing workflow

3.2 Modeling

Here, we use a subset (around 5,000 instances) which is obtained from the big data split into various X.list files. The first algorithm that was implemented is logistic regression, followed by Gaussian Naïve Bayes' algorithm, decision trees, random forest classifier, gradient boosting, artificial neural networks, and support vector machine. The parameters of the algorithms are kept as the default as obtained from the scikit-learn library available in python. The accuracies obtained are shown in Table 1.

As evident from Table 1, random forest classifier gives the highest accuracy, both for three classes and two classes. Also, it is easy to implement random forest classifier in distributed systems environment which will be useful when data are very large to analyze. Thus, the model obtained by it is considered along with its accuracy.

Table 1 Gross prediction accuracies by various classifiers

Algorithm	Three classes [0–0.5, 0.5–1, 1 +] (%)	Two classes [0–1, 1 +] (%)
Decision tree	60.62	76.37
Random forest	**67.72**	**78.74**
Gradient boosting	66.14	77.95
Gaussian Naïve Bayes	50.39	62.99
Logistic regression	56.69	59.05
Support vector machine (kernel = "linear")	53.54	62.20

4 Proposed Review Model

Now, the sentiment analysis part of our technique is focused that will aid the producers to improve their marketing strategies catering to the user requirements. The workflow diagram is shown in Fig. 5 (Figs. 3, 4).

Fig. 3 Review model preparation

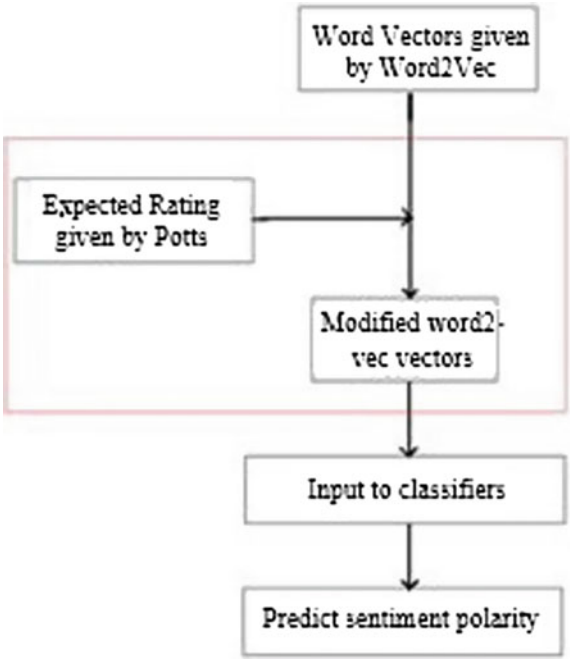

Fig. 4 Modified approach to word2vec

4.1 Data Collection and Preparation

The review corpus is taken from Large Movie Review Dataset, available from Stanford.edu [1]. It contains 25,000 labeled polar reviews for training and 25,000 for testing. Along with this, there is also unlabeled data available for use. This data is already preprocessed removing extra spaces and other delimiters. Emoticons are not considered in the model for now but they may be included in future work.

4.2 Vectorization

Here, the application of various techniques used for vectorization, including tf-idf, word2vec (both standard and modified), and doc2vec, is discussed.

For tf-idf, *25,000* rows are obtained in the co-occurrence matrix and the unique words in the training corpus will act as features. After removing most occurring and least occurring words (features) from the dataset, around *18,000* unique words are left, which will be the dimensionality for each row (document). Next, doc2vec is applied. The paragraph vectors are trained using 100 dimensions of the feature vectors. Different variants of doc2vec are used, i.e., the distributed bag-of-words (PV-DBOW), averaging of the word vectors and concatenation of the word vectors. The vectors are built using all training data, testing data, and unsupervised data as well. Now, these vectors are input to various classifiers, and the accuracies obtained are shown in Table 2. Followed by doc2vec, standard word2vec's two variants [3] are applied.

Proposed approach. Traditional model of tf and tf-idf does not capture the information about the context. Thus, they are in essence incapable of predicting any novice review even if it uses all the words from the trained model. This becomes even difficult when the review is written in a sarcastic tone. Also, word2vec algorithm can predict words pertaining to the particular topic. But for the sentiment classification, words in the same topic may be adhering to different emotions, i.e., both positive and negative. Also word2vec algorithm models vectors of words "good" and "great" nearby each other. But they have different intensities. This can be captured by incorporating the expected rating by Potts [9], in which negative words lie away from the positive ones as they have negative signed expected rating. Also, mod weights, i.e., expected ratings, are also taken into account.

This is implemented for both the variants of word2vec, i.e., continuous bag-of-words and skip-n-gram technique. The technique is delineated out in steps as follows:

- Obtain word vectors using word2vec and store them.
- Multiply each of the word vectors with the corresponding weights as obtained by Potts [9]. This multiplication is done in two ways:
  - First, just multiply the mod weights.
  - Second, multiply weights along with their polarity signs.
- Apply classification algorithms for sentiment classification using the modified vectors.

The accuracies obtained by using this technique shows great results with word2vec, giving accuracies comparable to doc2vec for Stanford dataset. Now, word2vec takes

less amount of space to store the vectors, and the time taken by classifiers is also considerably low. The running times are shown in Table 3 for reference. Also, the accuracies obtained by different classifiers are shown in Table 2, for both the variants of word2vec and even for tf-idf. Results obtained before and after the inclusion of expected rating are included for comparison. The accuracies have shown considerable improvement as shown.

In Table 3, it can be observed that training the vectors of words obtained using *tf* and *tf-idf* takes significantly large time to fit in the classifiers when compared to both *word2vec* and *doc2vec*. This is because of its dimensionality.

Also, doc2vec vector representations need paragraph vectors along with the word vectors whereas word2vec vectors are just word vectors. Thus, the space complexity is definitely reduced by using word2vec vectors in place of doc2vec vectors. Also, for predicting a new review's analysis, doc2vec needs to generate a paragraph vector unique to that review and then analyze its sentiment in contrast to word2vec vectors that just need vector representations of words, and analyzes the review by concatenating them. This even reduces the time required for sentiment analysis. Thus, word2vec is better than doc2vec in terms of space and time complexity. Our approach built on top of word2vec vectors gives two major advantages:

- It outperforms doc2vec in most cases and gives comparable accuracies in others.
- Running time is significantly low as compared to doc2vec, as we do not have to create a new paragraph vector to predict each of the new reviews, only word vectors are required for analysis, which is also depicted in Fig. 5.

For comparison, the same technique is used to classify reviews as positive and negative on Pang and Lee's dataset [11]. This dataset consists of 2000 reviews in all, with 1000 positive reviews and 1000 negative reviews. The ratings were determined from the star-ratings explicitly given by the users on IMDB while mentioning their reviews. For a 5-star rating system, reviews with 3.5 stars and above are considered positive reviews and 2 stars and below are considered negative reviews.

The accuracies obtained for some of the classifiers are shown in Table 4 for comparison. The execution time to specifically convert test documents to vectors is around 43.49 s for doc2vec for the initial phase and 1426.307 s for 10 iterations (performed so that the order of reviews does not affect the paragraph vectors) which is very high when compared to our proposed approach, which takes only 2.59 s for vectorization of Pang and Lee dataset.

Paragraph vectors in doc2vec technique are built on top of word vectors obtained by word2vec technique through incorporating context of the document. The paragraph vectors require a space complexity proportional to $O(n * |V|)$ where n is the number of documents (here review files) and $|V|$ is the size of the input vocabulary; whereas the space complexity for word vectors is proportional to $O(|V|)$. Our approach maintains the space complexity proportional to $O(|V|)$ as it is essentially a word vector representation. The time complexity of our modification applied on top of word2vec technique has time complexity linear to that of the size of the input vocabulary. And the time complexity of word2vec technique varies quadratically with the size of input vocabulary. Thus, the time complexity remains the same as word2vec, remaining less than doc2vec technique whose time complexity even depends on the number of the

Table 2 Accuracies for review analysis by various classifiers on different vectorization techniques (in %)

	tf	tf-idf	w2v (cbow)	w2v-mod (cbow)	w2v-pol (cbow)	w2v (sg)	w2v-mod (sg)	w2v-pol (sg)	d2v	d2v (bow)
RFC	78.05	77.62	74.18	83.08	84.98	78.05	84.75	86.33	73.34	71.81
DT	72.27	70.94	67.09	77.1	78.78	71.16	79.02	80.78	66.22	63.76
GBC	72.43	72.24	72.80	81.97	84.38	76.48	83.7	86.60	72.74	69.90
LR	85.39	88.09	85.29	88.09	88.55	86.43	83.53	87.48	86.24	88.62
SGD	83.73	87.8	80.18	87.98	88.58	85.94	86.91	87.34	79.44	86.46
LSVM	82.80	86.16	85.34	88.1	88.68	87.28	88.74	***89.01***	86.22	88.708

Table 3 Running times comparison of various vectorization techniques for sentiment polarity determination (in seconds)

	tf	tf-idf	W2v (cbow)	w2v- mod (cbow)	w2v-pol (cbow)	w2v(sg)	w2v-mod (sg)	w2v-pol(sg)	d2v	D2v (cbow)
Vectorization time	6.423	9.957	171.92	171.92 + 90	171.92 + 90	891.71	891.71 + 63	891.71 + 63	1837.6	5224.2
RFC	7.857	10.237	3.583	3.921	2.899	3.827	2.813	3.415	4.377	5.896
DT	33.72	49.02	4.780	5.784	6.324	5.724	5.421	6.454	6.606	7.675
GBC	0.979	3.126	2.212	1.599	1.578	1.474	1.425	1.595	2.263	2.368
LR	5.886	2.078	1.627	0.849	1.170	0.834	0.629	0.459	1.101	2.356
SGD	1.665	0.142	0.143	0.118	0.086	0.090	0.136	0.105	0.082	0.190
LSVM	5.160	0.628	14.244	6.087	4.514	1.040	0.682	***0.461***	16.330	13.157

RFC random forest classifier, *DT* decision tree, *GBC* gradient boosting classifier, *LR* logistic regression, *SGD* stochastic gradient descent, *LSVM* linear support vector machines, *w2v* word2vec, *d2v* doc2vec

Table 4 Accuracies comparison of doc2vec and our modification applied to word2vec (in percentage) for LSVM classifier

	Doc2vec	Proposed approach
Stanford dataset	88.708	**89.01**
Pang Lee dataset	86.75	**87.4**

input documents (here review files). Thus, our approach achieves both better time and space complexity as compared to doc2vec and still achieves better results. The F1 scores for precision and accuracy obtained for our modified approach are 0.88 and 0.896, respectively, for LSVM classifier on skip-n-gram variant combined with polar weights, which gives an F1 score of 0.89. The running times comparison for different techniques and two classifiers is depicted in Fig. 5, which shows considerably less running times for our modified approach as compared to both the variants of doc2vec.

Thus, after predicting the reviews polarity, the producers can figure out whether the movie is reaching out to the people in a positive way or not. Accordingly, producer can change his marketing strategy to improve the perception of people towards the movie and thus can expect better returns on the movie. This can be done in multiple phases before the release of the movie, so that at least producer can be assured that the movie will get a proper opening. Along with this, the distributor can also benefit by ordering more or less number of prints of the movie catering to people's requirements.

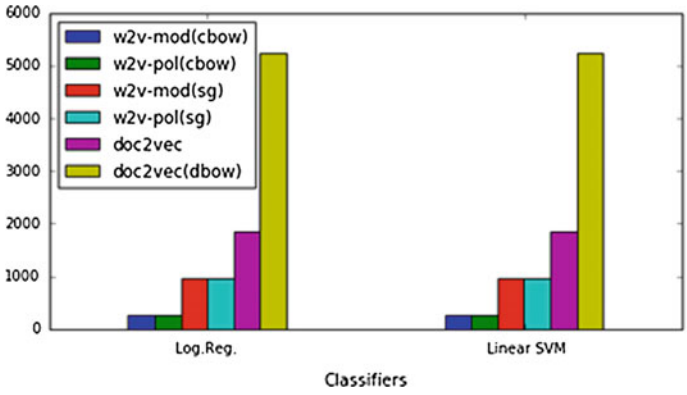

Fig. 5 Running times comparison for our approach and doc2vec for logistic regression and linear SVM classifiers

5 Conclusion and Future Work

The results for predicting the gross for the IMDB data are presented here. The best possible classifier, i.e., random forest classifier, is used for gross prediction. Then sentiment analysis on the user reviews is done and sentiment polarity classification is obtained using various classifiers, along with the accuracies provided for each of them, and their running times. Along with this, our approach is discussed, and accuracies obtained using that approach were comparable to doc2vec approach. Our technique is also tested on Pang and Lee's dataset, and the accuracies are presented. Better accuracy with less time complexity and less space complexity was obtained. It is tentatively concluded that producer can thus change his marketing strategies according to the user reviews and thus increase the profit for the movie and as a result the gross.

The future work includes integrating reviews from all social media possible, like Twitter, YouTube, etc. Also, the model can be extended to distributed systems.

References

1. Maas AL et al (2011) Learning word vectors for sentiment analysis. ACL
2. Rajaraman A, Ullman J (2011) Mining of massive datasets. Cambridge University Press, pp 1–17
3. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. ICLR Workshop
4. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: Proceedings of the 31st international conference on machine learning, PMLR 32(2):1188–1196
5. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning technique. In: Proceedings of EMNLP, pp 79–86
6. Han J, Kamber M, Pei J (2006) Data mining: concepts and techniques (The Morgan Kaufmann Series in Data Management Systems), 2nd edn. Morgan Kaufmann, San Francisco, CA, USA
7. Ng A (2016) Machine learning yearning: draft version 0.5
8. Malouf R (2002) A Comparison of algorithms for maximum entropy parameter estimation. In: Sixth conference on natural language learning (CoNLL), pp 49–55
9. Potts C (2011) On the negativity of negation
10. Sharma V, Rai S, Dev A (2012) Compr Study Artif Neural Netw 2(10)
11. Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd ACL, pp 271–278