**How Models and Document Types Affect Sentiment**

**Analysis Result**

Ziyao Sun; Zhuoyi Wu

The George Washington University

Data Mining

Professor Bellaachia

**Table of contents**:

[**Abstract**](#_d1994d4xh7fh) **3**

[**Introduction**](#_93d0v5t5klyb) **4**

[**Related Works**](#_v81keauvgxas) **5**

[**Graphics and Results**](#_iv7927fjcvmt) **6**

[**Learning Curves and Result (trained and tested with Sentiment 140 samples)**](#_gin5ioxwhwre) **6**

[**Histogram and result (trained and tested with IMDb comments)**](#_zaewofq0pdzu) **8**

[**Histogram and result(trained with IMDb comments and tested on Tweets/IMDb comments)**](#_d4rwz3t0osc) **9**

[**Histogram and result(trained with Tweets and tested on IMDb comments/Tweets)**](#_i2n7iomavzl4) **11**

[**Application**](#_4x8k7rv84wis) **12**

[**Conclusion**](#_x3u7gxbpptlj) **15**

[**References**](#_o3m9q82597th) **16**

## 

## **Abstract**

In our project, we dug in hotspots on Twitter and analyzed the sentiment of them. In this paper, our group will focus on two questions of sentiment analysis. First One is how well do different five algorithms(Vader(from NLTK); Naive Bayes (from NLTK); SVM; Random Forest; Neural Network) perform on the same dataset - sentiment 140[6]. Vader belongs to traditional analysis algorithms; Naive Bayes, SVM and Random Forest are machine learning methods; Neural Network is a deep learning method. With a maximum of 100,000 tweets sampling from sentiment 140 and 20,000 of them as a test dataset, the models’ accuracy scores are in the range of 51%-75%. The learning curves indicates increasing the training dataset may help improve the accuracy as the gap between test score and training score is large. Compared to all models tested, the Naive Bayes has the outperformance. Meanwhile, we used IMDb movie reviews to train and test the model. The models’ accuracy scores are in the range of 74%-90%. SVM and Neural Network have the best performance.

The second question is the model's generalization: whether the model we built based on one data works well on a different dataset. We did two cross tests: training models over IMDb movie reviews to predict the sentiment of tweets, and training models over Twitter data to predict the sentiment of IMDb movie reviews. In most cases, the accuracy scores of models drop in the interval of 50%-60%. However, the Naive Bayes model trained on tweets achieves the accuracy score of 72.3% while predicting the sentiment of IMDb movie reviews.

## 

## **Introduction**

Sentiment analysis has been used on many different area related to data mining, such as “social media monitoring, brand monitoring, the voice of the customer (VoC), customer service, and market research.”[1] People analyze those data’s sentiment in order to understand people’s standpoint(*polarity* e.g., positive, negative, neutral) and base on the opinions to make better business decisions.

In our project proposal, we plan to dig the hotspots in Twitter and analyze the sentiment tendency of hotspots. Digging the hotspots is not a big challenge; however, we are still questioning if current machine learning algorithms are reliable enough to do accurate sentiment analysis. This topic is directly related to our project, and is worth exploring. To answer this question, our group will focus on two parts of sentiment analysis.

Firstly, what are mainstream models used to do sentiment analysis nowadays, and how well do these algorithms perform. There are some traditional Machine learning methods, such as naïve Bayes, Logistic regression and Support vector machines. [2] And some deep learning methods, such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long-Short Memory networks (LSTMs). [2] In the final project of our machine learning course, we built and trained a Naive Bayes Model to analyze the sentiment of movie reviews from IMDb, by judging if the reviews are positive or negative. According to our test, our model can achieve an accuracy of about 85%. In this research paper, we plan to choose another mainstream model, maybe neural network which is very popular today, to test its performance in analyzing the sentiment tendency of hotspots, and then compare the neural network to Naive Bayes, figuring out which model can be more reliable.

Secondly, we will focus on how robust/general the machine learning methods are on text sentiment analysis. More specifically, will a machine learning model perform better on one type of text, but fails on another? Will our Naive Bayes model, which was trained on the data of movie reviews, also perform well in sentiment analysis of CNN’s news? We expect to do sentiment analysis on many textual materials. But the different material may not always be suitable to do sentiment analysis. For example, IMDB and customer feedback have stronger emotional directivity but more illegible sarcasm and informal languages (e.g., goooood, soooo bad…). On the other hand, news is written in written language, but weaker emotional signals. Therefore, how different topics, or different types of texts influence the robustness of machine learning models is also what we are concerned about.

## 

## 

## **Related Works**

Many researchers already excogitate on the subject of sentiment analysis. It is in the field of natural language processing that analyzes people's opinions, sentiments, evaluations, attitudes, and emotions via the computational treatment of subjectivity in text. In this paper, we are going to focus on opinion mining(positive and negative). We will briefly talk about some representative sentiment analysis models and canonical works related to those models.

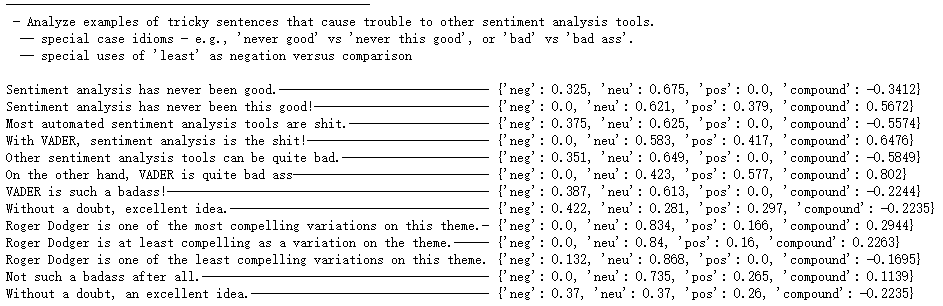
* Vader

Vader was published in AAAI in 2014. It is based on a gold-standard list of lexical features which are specifically attuned to sentiment in microblog-like contexts. Then it used embody grammatical and syntactical conventions for expressing and emphasizing sentiment polarity and intensity. They have discovered it can successfully assess the sentiment of tweets and it outperforms individual human raters.[3]

Beside the standard English words, it also take care of specific social media word:

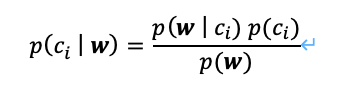
1. a full list of Western-style emoticons, for example, :-) denotes a smiley face and generally indicates positive sentiment[4]
2. sentiment-related acronyms and initialisms (e.g., LOL and WTF are both examples of sentiment-laden initialisms)[4]
3. commonly used slang with sentiment value (e.g., nah, meh and giggly).[4]

Some examples result by using built-in toolkit vander:



* Naive Bayes

Naive Bayes has been studied extensively since the 1960s and it was introduced into the text retrieval community in the early 1960s. Now, Naive Bayes classifier remains a popular methodology for Sentiment Analysis to judge sentiments as belonging to positive or negative. It’s a simple “probabilistic classifier” based on applying Bayes’ theorem with strong (naive) independence assumptions between the features.[5] Naive bayes model:



Where Ci is a class variable which can be positive or negative, **w** are evidence variables **w**=(W1, W2, …, Wn). Naive Bayes classifier assumes that evidence variables are independent. Theoretically, we will use the above formula to calculate the probability for each class given the evidence and then compare these two probabilities to determine which kind of sentiment they belong to.

* SVM (applied over doc2vec)

SVM model is one of the commonly used classifiers due to its robustness, high efficiency and high performance that outperform other popular classification models like Naïve Bayes or kNN (Hmeidi et al., 2008).[11] It uses a hyperplane to split data into two groups by increasing their dimensionality.

The classification effect also depends on the method of data preprocessing. The common methods of text vectorization include word2vec, doc2vec and TF-IDF. The classification result will vary if different models of text vectorization are chosen[11]. In this research, we use doc2vec for all models.

* Random Forest (applied over doc2vec)

The first algorithm for random decision forests was created by Tin Kam Ho[7]. An extension of the algorithm was developed by Leo Breiman and Adele Cutler[8], who registered “Random Forests” as a trademark (as of 2019, owned by Minitab, Inc.)[9] Random forests is composed by combining multiple decision trees and is a type of classifier which is very robust to noise and outliers because of randomness it provides. Random forests works as shown below:

Input: number of trees (A), training data (N), total features (F), subset of features(f)

Output: bagged class label for the input data

For each tree in forest A, select a bootstrap sample S of size N from training data, and then create the tree Tb by recursively repeating the following steps for each internal node of the tree: choose f at random from the F, select the best among f, split the node. Once trees are created, test instances will be passed to each tree and class label will be assigned based on the majority of votes[10].

* Neural Network (applied over doc2vec)

Finally, we try the Neural Network model, which is a great choice for performing word vectors with logistic regression. A neural network is a network or circuit of neurons, or in a modern sense, an artificial neural network, composed of artificial neurons or nodes. [12] The connections of the neuron are modeled as weights (positive values reflect excitatory connections and negative values reflect inhibitory connections) and all inputs are modified by a weight and summed. Fitting a neural network with early stopping and checkpoint can save the best performing weights on validation accuracy. Based on what we have observed during trials of different architectures with Doc2Vec document vectors, we can try to find the best performing architecture to fit a neural network model.

## **Graphics and Results**

### **Learning Curves and Result (trained and tested with Sentiment 140 samples)**

* Dataset Introduction

We Downloaded Sentiment 140 dataset from Kaggle which contains 1.6 million tweets fetched from Twitter, and about half of them are labeled as positive and another half of them are labeled as negative.(<https://www.kaggle.com/kazanova/sentiment140>). We sampled 100,000 tweets at random from Sentiment 140 and saved into our database. Then, we took 80,000 tweets from database at random (with random seed=10) as our training dataset, and the other 20,000 tweets as our test dataset. In our training dataset, we took the first 5000, 10,000, 20,000, 40,000 and 80,000 tweets to train and draw the learning curves.

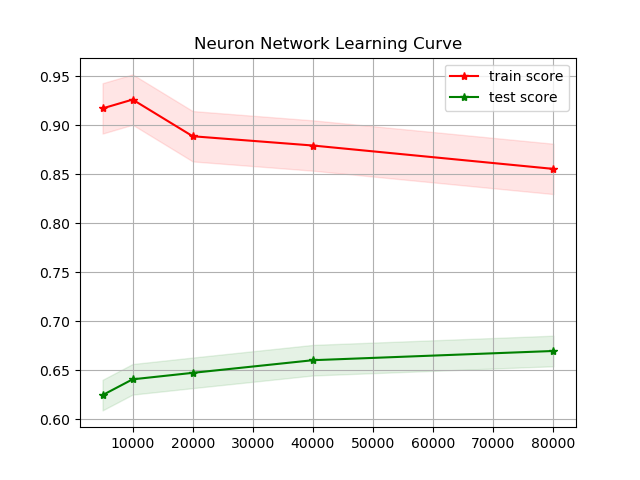
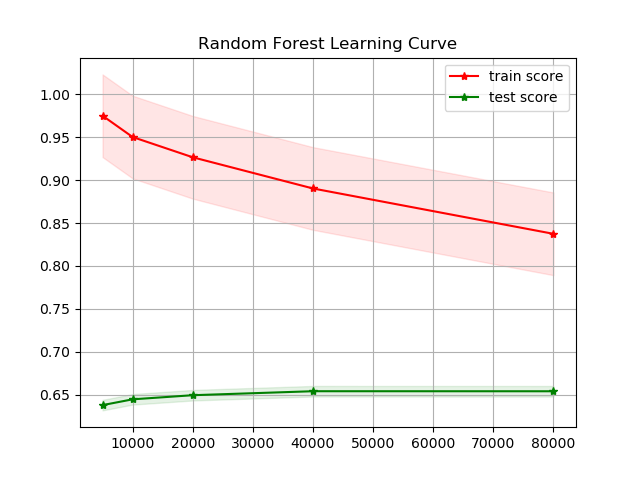
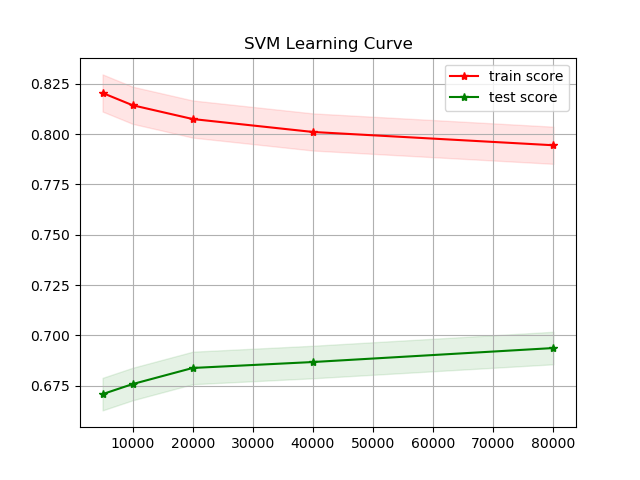
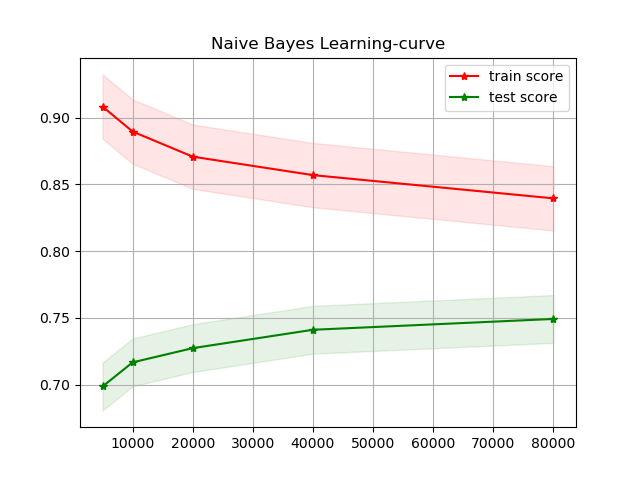
* Vader algorithm

Vader is rule-based, so it has no training data. The result includes positive, negative and neutral sentiment.Since Vader is rule-based and it has taken care of data preprocessing, we used raw data as its input data. The accuracy for 20,000 tweets from the test dataset is 0.51845.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Negative(pred) | Neutral(predict) | Positive(pred) |
| Nagative(real) | 4227 | 2636 | 3149 |
| Neutral(real) | 0 | 0 | 0 |
| Positive(real) | 1003 | 2843 | 6142 |

* Machine Learning Models

As introduced above, we used 20,000 tweets for testing (using the same test dataset in all machine learning models applied on Twitter sentiment analysis).



### **Histogram and result (trained and tested with IMDb comments)**

* Dataset Introduction

We wrote a crawler to fetch movie reviews from 250 top-rated movies and 100 lowest-rated movies on IMDb, and saved the data into our database. There are 9905 movie reviews fetched in total. Then, we took 7920 movie reviews at random with random seed=10, as our training dataset, and the other 1985 movie reviews as our test dataset.

According to *IMDb Movie Ratings Over the Years* by Nathaniel Johnston, “The average score (on a scale of 1 to 10) of those movies is 6.38 and the median score is 6.6”. Therefore, reviews with 6 points and below were divided into negative comments, reviews with 7 points and above are divided into positive comments.

In our training dataset, there are 4714 positive reviews and 3206 negative reviews.

In our test dataset, there are 1186 positive reviews and 799 negative reviews.

### 

### **Histogram and result(trained with IMDb comments and tested on Tweets/IMDb comments)**

#### 

### **Histogram and result(trained with Tweets and tested on IMDb comments/Tweets)**

## 

## **Application**

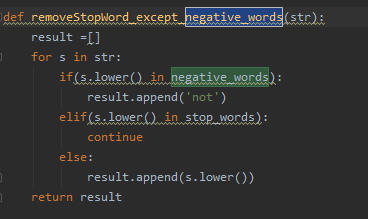
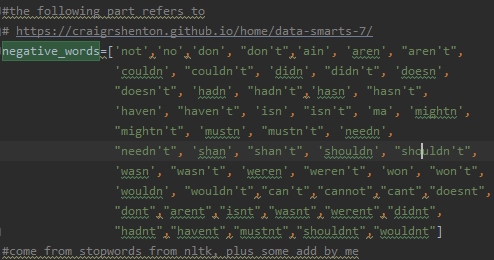
* Data Preprocessing
* Stop words

Nltk toolkit has its built-in stopwords corpus and it includes all languages. We only need english, so we modified the stopwords by removing all other languages, adding punctuation and some special words.



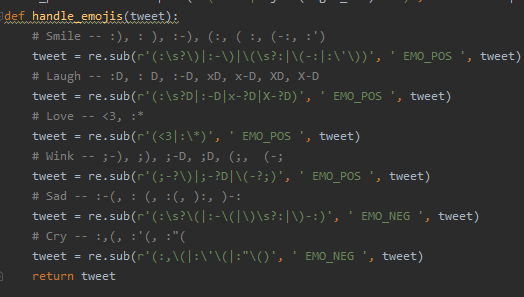
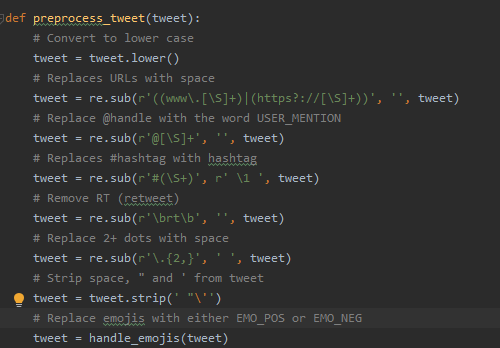
* Negative words

One of the improvements to the model is the handling of negative words. Negative words are important to sentiment analysis but they usually are filtered out by default NLTK. So we created a negative words list and replaced those negative words with ‘Not’ and kept them in processed texts.



* Spicial charactors

We remove URLs and @handle. And we replaces #hashtag with hashtag and emojis with either EMO\_POS or EMO\_NEG.

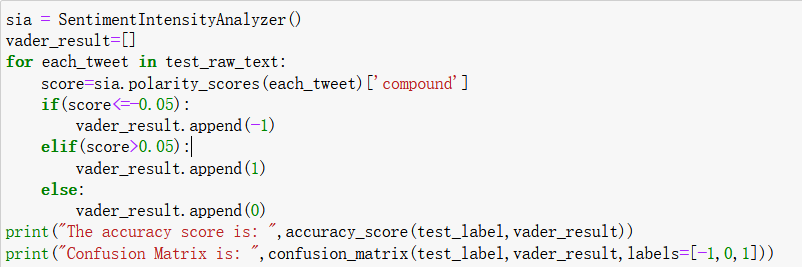


* Models

For SVM,RF and Neural Network, the training data and test data have been further processed with Doc2Vec.

* Vader

Built-in model calculated each sentence’s score. Positive score means positive and higher the score, the emotion is more positive. The built-in model considers sentences with score -0.05 to 0.05 as nature. But we removed the natural sentiment and only considered positive and negative.

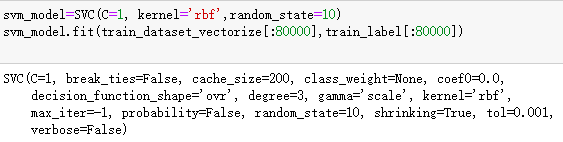


* Naive Bayes



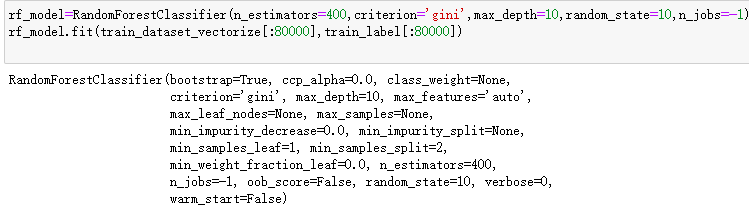
* SVM

The main hyper parameter：C=1,kernel=’rbf’,random\_state=10



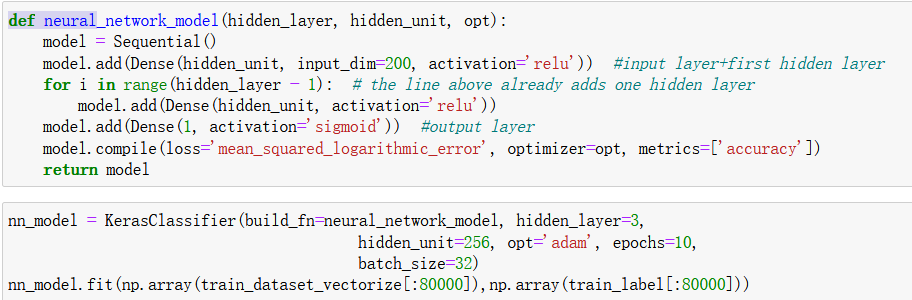
* Random Forest

The main hyper parameter：n-estimator=400, criterion='gini', max\_depth=10, random\_state=10.



* Neural Network

hyper parameter：hidden layer:3, each layer has 256 hidden units; each hidden layer uses ‘relu’ as activation function. output layer: use sigmoid as activation function. opt='adam', epochs=10, batch\_size=32



## **Conclusion**

Overall, the sentiment analysis models we used perform well on IMDb dataset,and perform much worse on Twitter dataset. Besides, our research proves that a sentiment analysis model trained and performing well on one dataset, may fail on another dataset.

The difference between IMDb movie reviews and tweets, is that IMDb movie reviews are much longer (most tweets are just one sentence) . Meanwhile, the topics and content of tweets are more diverse with more informal words (eg. okkkk, lmao, etc.) which also enhances the difficulty to predict the sentiment.

After analyzing our test result, we can conclude that for documents which have more words and obvious sentiment tendency, like IMDb movie reviews, machine learning models like Naive Bayes, Doc2Vec+Neural Network/SVM/Random Forest are good choices to predict sentiment (but notice that SVM has high time complexity, and training SVM takes much longer time especially the training dataset is large); Compared to rule-based models, machine learning models are better to predict the sentiment of positive and negative. However, for documents which have few words, Naive Bayes is more likely to achieve better performance. Meanwhile, more training data is needed to improve the performance of machine learning models to predict the sentiment of short texts. We will further investigate this problem in the future.

In addition, with the test result that Naive Bayes model trained on Twitter dataset achieves the accuracy score of 72.3% while predicting the sentiment of IMDb movie reviews, we may assume that Naive Bayes model trained over short documents could achieve much better robustness/generalization. More future work is needed to verify this assumption.

More code and data of our research is on <https://github.com/JoeyWu123/Text_Sentiment_Analysis_Research>

## 

## References

1. Sameer Balaganur, *10 POPULAR DATASETS FOR SENTIMENT ANALYSIS*. Feb 2020 <https://analyticsindiamag.com/10-popular-datasets-for-sentiment-analysis/>
2. Stanley Jose Komban, Raghavendra Prasad Narayan and Giorgio Orsi, *Deep Learning Models for Sentiment Analysis.* August 22, 2019, <https://underthehood.meltwater.com/blog/2019/08/22/deep-learning-models-for-sentiment-analysis/>
3. C. J. Hutto and Eric Gilbert, *VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text.* 2014-05-16 <https://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109>
4. <https://github.com/cjhutto/vaderSentiment#features-and-updates>
5. <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>
6. Go, A., Bhayani, R. and Huang, L., *Twitter sentiment classification using distant supervision.* CS224N Project Report, Stanford, 1(2009), p.12

<https://www.kaggle.com/kazanova/sentiment140>

1. Ho, Tin Kam (1995). [*Random Decision Forests*](https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf) *(PDF).* Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. *Archived from* [*the original*](http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf)*(PDF)* on 17 April 2016. Retrieved 5 June 2016.
2. [Breiman L](https://en.wikipedia.org/wiki/Leo_Breiman) (2001). *"Random Forests".* [*Machine Learning*](https://en.wikipedia.org/wiki/Machine_Learning_(journal))*.* 45 (1): 5–32.
3. U.S. trademark registration number 3185828, registered 2006/12/19.
4. Parmar, Hitesh and Bhanderi, Sanjay and Shah, Glory. *Sentiment Mining of Movie Reviews using Random Forest with Tuned Hyperparameters.* 2014.
5. Maria Mihaela TRUȘCĂ, *Efficiency of SVM classifier with Word2Vec and Doc2Vec models,* October 2019 <https://www.researchgate.net/publication/339261594_Efficiency_of_SVM_classifier_with_Word2Vec_and_Doc2Vec_models>
6. Hopfield, J. J. (1982). [*Neural networks and physical systems with emergent collective computational abilities*](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238)*.* Proc. Natl. Acad. Sci. U.S.A. 79 (8): 2554–2558. [Bibcode](https://en.wikipedia.org/wiki/Bibcode_(identifier)):[1982PNAS...79.2554H](https://ui.adsabs.harvard.edu/abs/1982PNAS...79.2554H). [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.1073/pnas.79.8.2554](https://doi.org/10.1073%2Fpnas.79.8.2554). [PMC](https://en.wikipedia.org/wiki/PMC_(identifier)) [346238](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC346238). [PMID](https://en.wikipedia.org/wiki/PMID_(identifier)) [6953413](https://pubmed.ncbi.nlm.nih.gov/6953413).
7. Nathaniel Johnston, *IMDb Movie Ratings Over the Years,* October 9th, 2009 <http://www.njohnston.ca/2009/10/imdb-movie-ratings-over-the-years/>
8. Ricky Kim, *Another Twitter sentiment analysis with Python — Part 10 (Neural Network with Doc2Vec/Word2Vec/GloVe),* Feb 9, 2018, https://towardsdatascience.com/another-twitter-sentiment-analysis-with-python-part-10-neural-network-with-a6441269aa3c