Word2Vec and SVM Fusion for Advanced Sentiment Analysis on Amazon Reviews

Haili Wang *

Tandon school of Engineering, New York University, New York 10012, the United States

* Corresponding Author Email: hw3255@nyu.edu

Abstract. As a matter of fact, sentiment analysis plays a crucial role in different kinds of reviews for online shopping. With this in mind, this study explores the application of Word2Vec and Support Vector Machine (SVM) in sentiment analysis of Amazon reviews. To be specific, initially, this study conducted preprocessing and feature extraction on a large-scale review dataset, revealing deep semantic relationships between words through the Word2Vec model. Subsequently, this study utilized SVM for model training and optimization, achieving efficient and accurate sentiment classification. According to the analysis, preliminary experimental results indicate that this combined method can effectively capture complex patterns and relationships in the text, demonstrating significant advantages in enhancing the accuracy as well as efficiency of sentiment analysis compared to traditional methods. Overall, these results shed light on providing a powerful tool for e-commerce platforms to understand better and analyze consumer opinions and feelings as well as guiding further exploration.

Keywords: SVM, Word2Vec, NLP.

1. Introduction

In recent years, the exponential growth in digital textual data, notably observed on platforms such as Weibo, has spurred a burgeoning interest in sentiment analysis [1]. Understanding the sentiments veiled in textual information not only augments online marketing strategies but also significantly advances the realms of customer relationship management and public opinion monitoring. The essence of sentiment analysis hinges on distinguishing the emotional undertones of texts into categories such as positive, negative, or neutral, thereby fostering a rich platform for data-driven insights and strategies. Historically, sentiment analysis relied heavily on labor-intensive feature engineering, which necessitated the availability of large annotated corpora [2]. However, with the inception of Word2Vec, a tool adept at automatic feature extraction, the process has been notably simplified and improved. Initially applied to short texts, Word2Vec has gradually demonstrated its prowess in the sentiment analysis of citations as well [2]. This tool, developed by Mikolov et al. in 2013, transforms words into vectors, facilitating the representation of semantic relationships in a multidimensional space [3, 4]. It has found profound applications in various languages, including Bengali, where it aids in sentiment classification by drawing upon the semantic closeness of words in vector space [5]. Several studies have ventured to enhance the efficacy of sentiment analysis through innovative methodologies. Alshari et al. explored the clustering of Word2Vec features to construct a feature set for sentiment analysis, demonstrating promising outcomes in terms of efficacy compared to baseline approaches [3]. Moreover, a concerted effort has been made to augment the sentiment lexical dictionary to enhance sentiment analysis performance, showing promising results in early evaluations [4, 6].

Building upon the robust foundation laid by previous studies, this research seeks to further the horizon of sentiment analysis, especially in the e-commerce landscape. Leveraging the capabilities of Word2Vec and SVM, this paper introduces an enriched sentiment analysis mechanism that integrates semantic analysis with advanced classification techniques. The objective is to unveil nuanced textual patterns and sentiments in customer reviews, facilitating data-driven decision-making in the e-commerce sector. By amalgamating existing methodologies' strengths, this paper proposes a more refined, accurate, and efficient sentiment analysis framework [7-11].

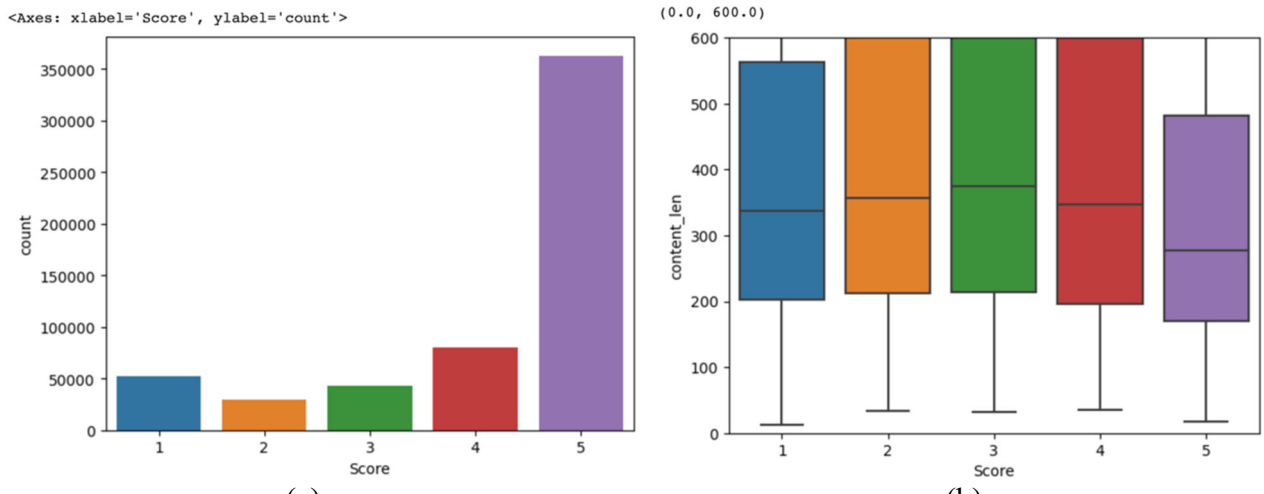
The forthcoming sections of this paper are systematically arranged to provide an extensive understanding of the presented study. After this introductory segment, next section details the implemented methodology, highlighting the fusion of Word2Vec and SVM. Result unveils a thorough examination of the experimental results, showcasing the improved ability in identifying complex patterns within texts. In conclusion, it summarizes the insights gained from the investigation, suggesting potential directions for subsequent research in this area.

2. Methodology

The methodology's core hinges on integrating the Word2Vec model to extract textual features and utilizing a Support Vector Machine (SVM) for classification tasks. The model's performance is then evaluated with the AUC metric, which analyses the trade-off between a valid positive rate and a false positive rate, providing a comprehensive understanding of the model's proficiency in distinguishing complex textual patterns. For this project, the Amazon Fine Food Reviews dataset, accessible through Kaggle, is utilized, where a comprehensive examination of the dataset will be undertaken to understand the overarching trends and disparities between positive and negative reviews present within the dataset [7].

2.1. Data Overview

This study is utilizing the Amazon Fine Food Reviews dataset available on Kaggle for the experiment. This dataset comprises reviews of fine foods from Amazon, spanning over ten years (from October 1999 to October 2012), encompassing approximately 568,454 reviews given by 256,059 users towards 74,258 products. The dataset includes detailed user and product information, ratings, and plain text reviews. The data files available are in two formats: a CSV file and an SQLite database containing a table named 'Reviews.' The focus will be on users with more than 50 reviews, summing up to 260 users. Through the initial observation, this study identified missing values in the "Profile Name" and "Summary" fields. However, as these fields do not significantly influence the analysis, this study will not address these missing values in the preprocessing phase. Initially, a comprehensive observation of the dataset will be conducted to grasp the overall state of the dataset, including the general distribution of ratings and the balance between positive and negative reviews. The reviews' length and respective ratings will be explored to discern patterns in users' feedback intensity. This analysis revealed an imbalance in the rating distribution, with more positive reviews than negative ones. Moreover, it was observed that 1-star reviews tend to be longer, suggesting that dissatisfied customers are providing more detailed feedback. The results are shown in Fig. 1.



(a) (b)

Figure 1. Text rating distribution (a) and boxplot of review length and rating (b).

This section will outline the methodology for building a sentiment analysis model based on Amazon reviews with ratings ranging from 1 to 5. Given the ratings' limited discriminatory power for sentiment analysis, this study proposes a re-labeling scheme to enhance the efficiency of the sentiment classification model. Here is a step-by-step breakdown of the methodology. To build a more efficient sentiment analysis model, one will categorize the reviews into two distinct classes: positive and negative. This categorization will be based on the existing Amazon ratings, wherein:

- Reviews rated 4 to 5 will be labeled positive, indicating customer satisfaction.
- Reviews rated 1 to 3 will be considered negative reviews, highlighting dissatisfaction.

Before moving forward with model building, one needs to convert the text data into a format that can be fed into the model. The following steps will be undertaken:

- Text Preprocessing. Initially, this study will preprocess the reviews by removing irrelevant information, such as special characters and stopwords, and performing stemming or lemmatization to reduce words to their base form.
- Text Vectorization. Subsequently, this paper will transform the cleaned text data into a sparse matrix, which will serve as the input for the model. This transformation will be done using various techniques such as Bag-of-Words or TF-IDF (Term Frequency-Inverse Document Frequency).
- Feature Selection with TF-IDF. Particularly this study will employ the TF-IDF methodology for this project. The TF-IDF technique will help weigh the words according to their importance in the document while also considering their frequency across all documents.

2.2. Word Embedding: Word2Vec

This study predominantly focuses on utilizing the Word2Vec model to extract textual features from the reviews. Before diving into Word2Vec, it is essential to comprehend the basic principles of Natural Language Processing (NLP). In NLP, the finest granularity is words, which formulate sentences, paragraphs, and documents. It is critical to transform words into a computer-able format to facilitate machines in understanding human language. The transformation process from symbolic representation (words in various languages) to numerical model is of paramount importance. Word embeddings, also known as word vectors, are a category of methodologies to map words to a vector space, potentially encapsulating semantic relationships among words. Initially, simple one-hot encodings were used to achieve this transformation; however, they lack semantic information and result in high-dimensional vectors. To overcome these limitations, the Word2Vec model, introduced by the Google research team in 2013, offered an efficient approach to computing word vectors that reduced dimensionality and encapsulated semantic relationships among words. Word2Vec hinges on the idea that words appearing in similar contexts should have identical word vectors (seen from Fig. 2). This idea has been implemented in CBOW (Continuous Bag of Words) and Skip-gram. The CBOW method predicts the target word based on its context, whereas the Skip-gram method predicts the context based on the target word. In the CBOW model, the focus is on predicting a target word based on its surrounding context words. The steps involved in the CBOW method are as follows:

1. Initialization: Context words are represented using one-hot encoding, forming the input layer where vocabulary size is denoted as V, and the number of context words as C.
2. Hidden Layer Processing: Each one-hot encoded vector is multiplied by the weight matrix W, which connects the input layer to the hidden layer.
3. Vector Averaging: The vectors obtained from step 2 are averaged to form a single vector at the hidden layer.
4. Output Layer Processing: This single vector is then multiplied with another weight matrix W', transitioning from the hidden layer to the output layer.
5. Softmax Activation & Prediction: The output layer undergoes a softmax activation, resulting in a probability distribution over the V-dimensional vocabulary. The highest probability index is chosen as the predicted target word.

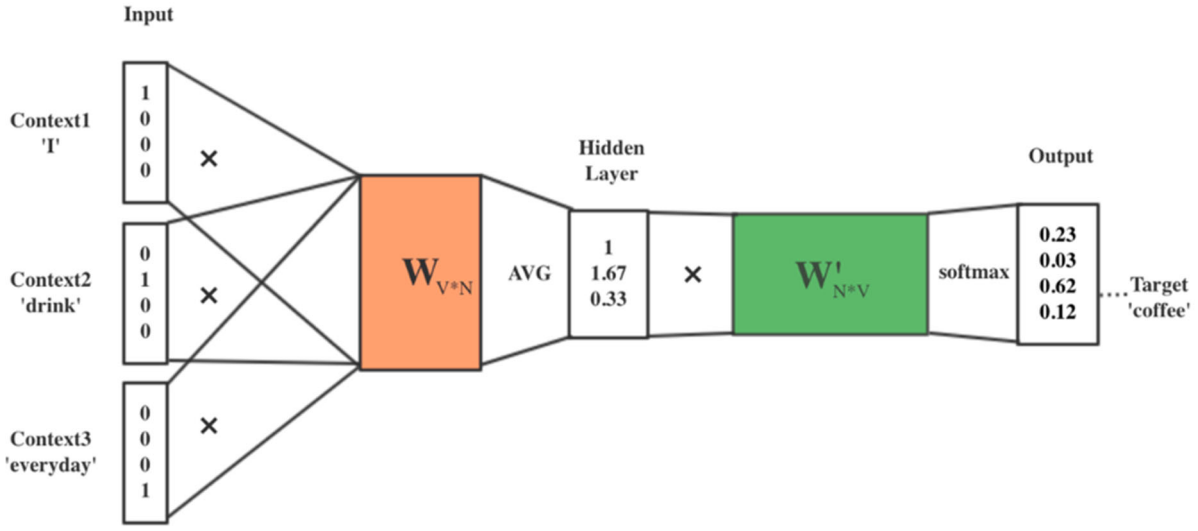


Figure 2. The whole process of CBOW.

Conversely, the Skip-gram method operates inversely to the CBOW, focusing on predicting the surrounding context words given a target word. Despite the inversion in operational steps, the architectural nuances and conceptual understanding remain parallel to the CBOW model. This study initiates from the target word and progress backward through the CBOW steps to arrive at the context words. A visualization of this process mirrors the CBOW process diagram, only read from right to left. In the experimentation, this study aims to explore the potential of the Word2Vec model in identifying optimal features for the analysis. The experiment with different dimensions for the word vectors, specifically setting dimensions at 100 and 200 to find optimal results. The model's effectiveness will be evaluated using the AUC metric, defined as the area under the ROC (Receiver Operating Characteristic) curve. The ROC curve, plotted using a valid positive rate (sensitivity) and false positive rate (1-specificity), is a robust tool to understand the model's proficiency in distinguishing between the positive and negative reviews in the dataset. Through this method, one aims to craft a model that can accurately and effectively analyze the intricate patterns within the textual data available in the Amazon Fine Food Reviews dataset.

2.3. Support Vector Machine

In the initial phase of the methodology, each extracted feature and its corresponding class label - such as positive and negative - are stored in a database. These labels serve as the ground truth during the SVM training and evaluation phases. The Support Vector Machine (SVM), primarily a binary classifier, is trained using these labeled features to differentiate between various classes of emotions. Despite its binary nature, it can categorize multiple categories, mapping each element to its respective class label. The SVM employs a non-linear mapping, denoted as $\Phi(x)$, which projects the training set into a higher-dimensional space, transforming the non-linear problem into one that can be linearly separated. The optimal separating hyperplane is described by the equation:

$$y = \omega^T \Phi(x) + b \tag{1}$$

Here, ω and b represent the weight vector and bias vector of the SVM, respectively. To find the optimal ω and b, a relaxation factor ξ_i is introduced to transform Eq. (1), resulting in a secondary optimization problem characterized by a penalty parameter, C, and the relaxation factors. Further, the problem is transformed using Lagrange multipliers, culminating in the computation formula for the weight vector ω as:

$$\omega = \sum \alpha i y i \Phi(xi) \cdot \Phi(x) \tag{2}$$

The SVM classifies data using a decision function, given as:

$$f(x) = sgn(\sum \alpha iyi\Phi(xi) \cdot \Phi(xj) + b) \quad (3)$$

To alleviate computational complexity, a kernel function $k(x, x_i)$ substitutes the dot products $\Phi(x_i) \cdot \Phi(x_j)$, modifying Eq. (3) to:

$$f(x) = sgn(\sum \alpha iyik(x, xi) + b) \quad (4)$$

Among various potential kernel functions, the Radial Basis Function (RBF) is selected owing to its favorable generalization ability and fewer parameters. The RBF kernel nonlinearly maps samples into a higher-dimensional space, accommodating non-linear relationships between class labels and attributes more proficiently compared to polynomial kernels and offering less numerical difficulty. The decision function thus transforms to:

$$f(x) = sgn(\sum \alpha iyik(x, xi; \sigma) + b) \quad (5)$$

Here, σ represents the width parameter of the RBF kernel, controlling the separation margin of the classes.

Table 1. Scores of Bi-LSTM Model.

Model	Vocab_dim	maxlen	Score
Bi-LSTM	100	100	0.6486
		160	0.6552
		200	0.6543
	200	100	0.6502
		160	0.6789
		200	0.8108

3. Result & discussion

This experiment implemented and analyzed four classification algorithms: Bi-LSTM, TextCNN, CNN+Bi-LSTM, and Support Vector Machine (SVM). Table 1 summarizes the findings and optimal parameter settings during the experimentation. When the word vector dimension was set at 200, and the maximum sentence length was capped at 200, the Bi-LSTM model exhibited the most promising performance. In conditions diverging from these settings, the variations in model performance were not significantly distinct. The TextCNN model achieved optimal results when the word vector dimension was 100 and the maximum sentence length was restricted to 160. Notably, a word vector dimension of 100 proved superior to an extent of 200, facilitating reduced computational demands and shorter runtimes. Furthermore, when the word vector dimension was configured at 100 with a maximum sentence length of 200, the TextCNN model again reached its peak performance. The results are shown in Table 2.

Table 2. Scores of TextCNN Model.

Model	Vocab_dim	maxlen	Score
TextCNN	100	100	0.6321
		160	0.8203
		200	0.8108
	200	100	0.6511
		160	0.8112
		200	0.8103

During the feature extraction process, TF-IDF demonstrated a higher efficacy than the word2vec method. Additionally, removing punctuation marks positively impacted the model's accuracy, enhancing its predictive prowess (seen from Table 3). The SVM model's optimization was mainly

focused on tuning three critical parameters: the penalty parameter (C), the type of kernel function, and the kernel coefficient (gamma). The optimal parameter tuning was facilitated using the GridSearchCV method, a utility provided by the Python package sci-kit-learn. Through this approach, the experiment determined the best parameter set for the SVM model to be {'C': 4, 'gamma': 0.25, 'kernel': 'rbf'} as given in Table 4.

Table 3. Scores of CNN+Bi-LSTM Model.

Model	Vocab_dim	maxlen	Score
CNN+Bi-LSTM	100	100	0.7962
		160	0.7856
		200	0.8465
	200	100	0.8112
		160	0.8375
		200	0.8198

Table 4. Scores of SVM Model.

Model	Feature_extraction	Remove_punctuation	Score
SVM	TF-IDF	Remove	0.8967
	Word2vec	Remove	0.8345
		Don't remove	0.8478

A set of comparative tests was undertaken to evaluate the effectiveness of various feature extraction methods, specifically comparing TF-IDF and word2vec and examining the effect of punctuation marks on the model's precision. To guarantee a comprehensive assessment, a 20-fold cross-validation was employed. This stringent validation technique enabled an in-depth analysis of the disparate models and their respective setups (seen from Fig. 3).

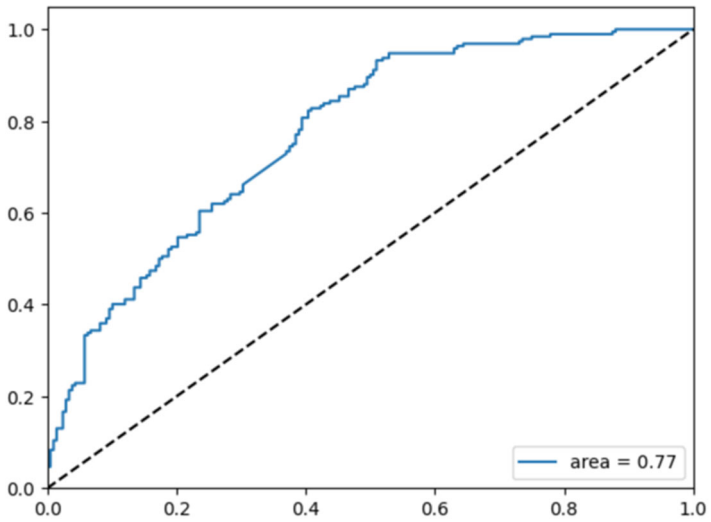


Figure 3. The result of SVM- TF-IDF model.

4. Conclusion

This study has garnered several significant insights regarding the effectiveness of various classification models and feature extraction techniques on the Amazon Fine Food Reviews dataset. The findings suggest optimal strategies for more accurate and efficient sentiment analysis applications in future projects. One notable outcome of the experiment is the superior performance of the Support Vector Machine (SVM) classifier for this particular task. It demonstrated higher classification efficacy than CNN+Bi-LSTM, TextCNN, and Bi-LSTM models, and its training time was significantly shorter compared to the other deep learning models utilized in the study. This

reiterates the SVM's potential as a powerful tool for sentiment analysis, combining both efficiency and accuracy.

Furthermore, the meticulous parameter-tuning process unearthed critical feature extraction and representation insights. It was observed that the dimensionality of the word vectors created using Word2Vec did not substantially impact the final classification performance. In contrast, the choice of maximum sentence length played a more decisive role. This can be attributed to a word vector dimension of 100 being sufficiently representative, and further increasing the measurement does not induce significant variations. On the other hand, modifying the maximum sentence length alters the degree of zero-padding in shorter sentences and the amount of information truncated in longer sentences, thereby having a more pronounced influence on the comprehensive representation of a sentence. Moreover, traditional methods of text feature extraction surprisingly surpassed the neural network-based Word2Vec model in terms of classification performance. DESPITE BEING COMPUTATIONALLY MORE INTENSIVE, employing TF-IDF for sentence vector representation managed to encapsulate a more holistic picture of the sentence semantics, overshadowing the performance of Word2Vec, which offered a comparatively diminished expression. Lastly, an essential revelation during data preprocessing was the positive impact of punctuation removal on the final classification performance. Omitting punctuation marks enhanced the model's ability to discern and classify sentiments more accurately, elevating the overall model performance. In reflection, this study underscores the necessity to judiciously choose the classification model and feature extraction techniques to ensure a well-rounded and effective sentiment analysis model. These insights will foster further innovation and development in the field, driving towards more accurate and efficient solutions for text data mining and sentiment analysis.

References

[1] B. Xue, C. Fu and S. Zhan, 2014 IEEE International Congress on Big Data (2014) pp. 358-363.

[2] H. Liu, arXiv preprint arXiv: 1704. 00177 (2017).

[3] E. M. Alshari, A. Azman, S. Doraisamy, et al., 2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP, 2018) pp. 1 - 5.

[4] S. Thavareesan, and S. Mahesan, 2020 Moratuwa Engineering Research Conference (MERCon, 2020) pp. 272 - 276.

[5] M. Al-Amin, M. S. Islam and S. D. Uzzal, 2017 International Conference on Electrical, Computer and Communication Engineering (ECCE, 2017) pp. 186 - 190.

[6] B. Xue, C. Fu and S. Zhan, 2014 IEEE International Congress on Big Data (2014), pp. 358 - 363.

[7] A. Wahal, M. Aggarwal and T. Poongodi, 2022 3rd International Conference on Intelligent Engineering and Management (ICIEM, 2022), pp. 484 - 489).

[8] Y. Li, K. Bontcheva and H. Cunningham, Natural Language Engineering, 15 (2), 241 - 271 (2009).

[9] K. P. P. Shein and T. T. S. Nyunt, 2010 Second International Conference on Communication Software and Networks (2010) pp. 169 - 172.

[10] S. M. Abu Taher, K. Afsana Akhter, and K. M. Azharul Hasan, 2018 International Conference on Bangla Speech and Language Processing (ICBSLP, 2018), pp. 1 - 5.

[11] J. C. Ferrão, F. Janela, M. D. Oliveira and H. M. G. Martins, 2013 IEEE International Conference on Healthcare Informatics (2013), pp. 511 - 516).