A Convolutional Neural Network for Amazon Customers' Reviews Rating Classification

1056369

ABSTRACT

The rapid growth of online shopping has caused to attract many attentions in this domain recently. Three beneficiaries are involved in this issue (suppliers, retailers, and consumers). A critical issue for all these parties is customers' reviews of products that can help them make a better decision. To answer the questions of this domain, there are diverse studies that use text mining techniques. In this study, we want to answer some questions such as 'how we can predict review ratings just by focusing on review titles and texts?'. We scrutinized this issue by using the Convolutional Neural Network (CNN) algorithm and could get a result with acceptable accuracy in comparison to other methods.

1 INTRODUCTION

The advent of online shopping has revolutionized the productivity of the supply chain on both sides, demand, and supply. In this respect, one of the most significant criteria is the consumer reviews and ratings for a specific product. The consumer opinions could be valuable for all manufacturers, suppliers, sellers, and buyers. Based on this beneficial information, manufacturers would try to launch and re-launch their new products, and suppliers would be able to set their orders more productively. Besides, sellers could focus on more effective distributing channels, and eventually, buyers might purchase their merchandise more confidently. In this regard, decision making and interpreting based on tons of viewpoints of users are almost impossible for a person or even companies, so this matter signifies more the importance of Data Mining. In this project, we apply both popular machine learning and deep neural network algorithms to predict ratings of a product, oriented by title and text reviews. If the system can predict the score of products from users' text opinions, it could propose an automatic suggestion while the customers are expressing their opinion. This technique makes online shopping websites more convenient and user-friendlier. There are also several other applications for the proposed model such as spam scoring detection or fake scoring detection to evaluate the degree of websites' reliability. The idea of this project is established based on answering some questions that are followed by this. Do deep neural networks perform better than traditional machine learning algorithms in opinion mining? Is it possible to have a three-dimension representation of words? Will I have a more accurate optimizer by switching some thresholds and parameters in predicting rating from users' text reviews? Can we have a better result if we give weighting to title more than text of reviews? It should be mentioned that my work is available via the Github [1] URL.

2 RELATED WORK

There is a wide range of research papers associated with product reviews and opinion mining. Tan, W. et al. showed how both traditional and deep neural machine learning methods could classify the reviews to detect fraud scorings [1]. They studied some types of KNN, SVM, and Naïve Bayes as traditional methods also checked the performance of LSTM as a deep neural method. Accordingly, they revealed that neural models classify better than traditional ones. Additionally, they answered some fundamental questions like 'why enlarging of feature space not necessarily leads to having a better performance of the model?' In another effort, Xing F. et al. focused on spam detection in online customer's reviews of Amazon to answer this question of whether Amazon's reviews are spam or not [2]. They concentrated on this issue by using traditional methods like SVM, Naïve Bayes, and Random Forest. Based on semantic analysis of reviews, they promisingly recognized that the Amazon reviews are reliable, so ratings are not spam. Machine learning has witnessed a remarkable revolution with the rise of deep learning algorithms over the recent decade. Afterward, General-Purpose Computing on Graphics Processing Units (GPGPU) has simplified by the expansion of Convolutional neural networks and the spread of parallel computing libraries. To allocate shorter codes to the most commonly used characters, Marinho, W. et al. proposed a novel text-to-tensor representation that is based on information compression techniques [3]. This novelty is language-independent without any requirement to do pretraining. Besides, it generates an encoding without any information loss. They concluded that the number of parameters to be optimized decreased considerably. Also, by a one-hot encoding method for representation (which we used in our work), we can have a more reasonable classification performance. Rozi M. F. et al. have research in which they could conclude the state-of-the-art accuracy for training data performance [4]. They applied CNN for feature extracting and L2-SVM for classification. Moreover, They indicated how deep neural networks are able to work well with a few number iteration to have a terrific opinion mining and classify reviews.

3 DATASET AND FEATURES

3.1 Data set description

Consumer Reviews of Amazon Products [6] is a data set that has considered to use for this project. Thanks to Kaggle's repository, getting access to this data set is publicly possible via the proposed link [2]. The dataset was donated in 2017 and the last updated version was released in 2019. Besides, the set includes 34660 consumer reviews for some of Amazon's products like the Kindle and Fire TV Stick. The dataset contains basic product information listed as rating, text and title of the reviews, manufacturer, brand, and other information.

[1] https://github.com/rezashokrzad/Text-Mining-Final-Project.git

[2] https://www.kaggle.com/datafiniti/consumer-reviews-of-amazon-products

TxMM'20-'21, Radboud University, Nijmegen, Netherlands

3.2 Experimental Setup

To have a standard experimental setup, we implemented Python 3.8 as a language programming and Google Colab as a development environment. Besides, TensorFlow 1.x and nltk were the most substantial libraries to do this study. Since in this study we want to use CNNs (Convolutional Neural Networks), TensorFlow is an easy-to-use framework to train CNN. Furthermore, nltk is an NLP (Natural Language Processing) library by which we can use several useful functions to pre-process texts. Also, we Use *gensim* to load a word2vec model *pretrained* on google news and perform some simple actions with the word vectors.

3.3 Data Preprocessing

After going through the dataset, first, we found a considerable difference in the number of labels which shows in Figure 1. To address the imbalanced data, we used a popular method within it we merge some classes together. In this attempt, we combined four ratings from 1 to 4 to class 0 and only rating 5 to class 1 that Figure 2 shows the result. After merging we have two high-rate and low-rate classes that are more balanced. Although these classes are still relatively unbalanced, merging is just out only found solution to tackle the problem. Accordingly, Yang Z. et al. have done a bias anchoring research to explain why people irrationally tent to give a higher rating which is not necessarily their real feeling [5]. Besides, we realized that around 33 data points have not any rating when we moved through the data. Since the number of missing data was negligible, to handle this issue, we allocated them to class 1.

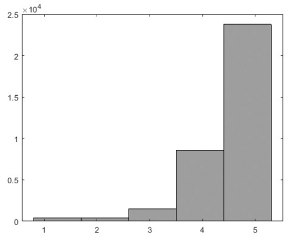

Figure 1: Ratings Distribution Histogram

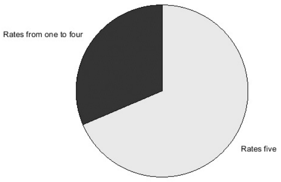

Figure 2: Resampling Ratings Result Pie Chart

Additionally, to apply data more productively, first, most of the columns with marginal information are eliminated. Followingly, merely three columns that are rating, title, and text of reviews, in a respect system, are extracted. Afterward, Figure 3 illustrates the word clouds of the text reviews related to classes 1 to 5 that give us a better perception of differences among the reviews. Looking at the clouds clarifies that classification is relatively sophisticated because there is no distinct difference walking through the classes. In Natural Language Processing (NLP), there are some fundamental steps to prepare the text. eliminating stopwords, making all words in lower case, stemming the rest of the words, and tokenizing to have separated words are the most remarkable steps to create our corpus. In this case, our stemmer was PorterStemmer. Figure 4 depicts the mentioned process for review number 30 that is chosen randomly. Besides, we used a proper pre-trained model from the Gensim library, named word2vec-google-news-300, that has been trained by news of Google, so it looks valid sufficient for word embedding. After all, we made a three-dimensional representation 30×30×1 that can be seen in Figure 5 to prepare data as input of CNN. This step has been explained in detail in the next sections. mention that our activation function is RELU function.

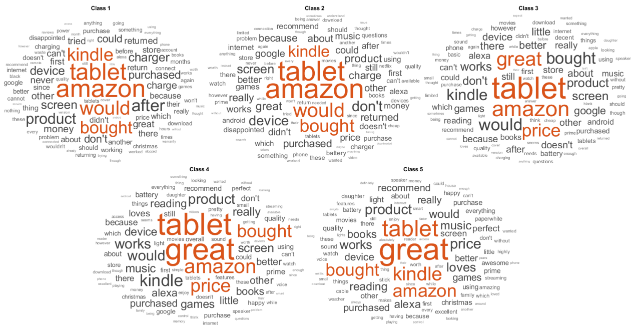

Figure 3: The word clouds of reviews' ratings

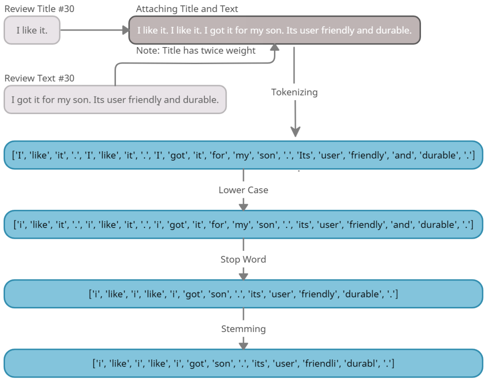
Review Title #30
I like it.
Attaching Title and Text
I like it. I like it. I got it for my son. Its user friendly and durable.
Note: Title has twice weight
Review Text #30
I got it for my son. Its user friendly and durable.
Tokenizing
['I', 'like', 'it', '.', 'I', 'like', 'it', '.', 'I', 'got', 'it', 'for', 'my', 'son', '.', 'Its', 'user', 'friendly', 'and', 'durable', '.']
Lower Case
['i', 'like', 'it', '.', 'i', 'like', 'it', '.', 'i', 'got', 'it', 'for', 'my', 'son', '.', 'its', 'user', 'friendly', 'and', 'durable', '.']
Stop Word
['i', 'like', 'i', 'like', 'i', 'got', 'son', '.', 'its', 'user', 'friendly', 'durable', '.']
Stemming
['i', 'like', 'i', 'like', 'i', 'got', 'son', '.', 'its', 'user', 'friendli', 'durabl', '.']

Figure 4: NLP Preprocessing For a Random Review

3.4 Feature Engineering

To use CNNs (Convolutional Neural Networks), first, we needed to look at the text input of CNN as a picture with three-dimension 30×30×1 (the first dimension refers to the number of word2vec representation, the second one refers to the number of tokens to create a sentence, and the third one related to RGB in picture that we just assigned 1). Additionally, to construct a fielded dictionary,

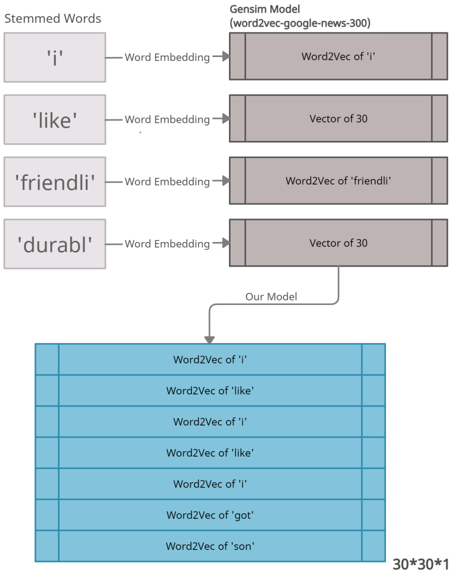

Figure 5: Creating Three-dimension Document

my approach was weighting to review titles two times bigger than review texts.

4 METHODOLOGY

The main objective of this study is predicting ratings from the title and text of reviews. To meet this purpose, we needed a classifier to label reviews. In this study, we concentrate on CNN as our classifier. In following subsections, there are a detailed explanation about the main approach (CNN) and comparing it to three other classifiers named Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Decision Tree (DT). These models were trained by two approaches of word and document representation (word2vec and tf-idf). In this study, we had two relatively innovative setups which are giving double weighting to title reviews and single weighting text reviews (fielded approach) and creating a two-dimension representation. By comparing the outcome, we could gain a better perception of how these algorithms work and what their advantages and drawbacks are. In the fielded approach, through the operation of concatenating titles and texts to have a single sentence for each review, we duplicated the title to give it a double effect in the representation. Since we assumed titles of reviews are a decent substitute for texts to guess ratings, we added more weight to titles than the text reviews by duplicating them at the first of the representation. This approach can be useful if an online retailer wants to make reviews semi-automated, for instance, while the user is making a title, the system can offer simultaneously a suggested rating based on the typing title. In the case of CNN's input, we need to have a three-dimensional picture (matrix). In this regard, first, we create word2vec of each word in 30 digits, then make a matrix with 30 cells of a sentence's words. To complete the process, we should consider the third dimension just 1, because our data is not a picture with RGB.

4.1 Proposed Method

A Convolutional Neural Network (CNN) is a class in deep learning that most typically utilized to analyzing visual imagery. CNNs are normalized editions of Multilayer Perceptron (MLP), so their design can be appropriate for processing two-dimension data. Since they are deep neural networks, they can accept a large amount of data, so in our work we initial feed model by three-dimension matrix which is shown in figure 5 as the output. To be more specific, two main layers named convolution and pooling layers are the foundation of CNN architecture. Figure 4 shows the architecture of the feed-forward process of CNN that is a decent explanation of how the two mentioned layers work [3]. This operation is the first step of building the TensorFlow graph and this graph will be completed by backpropagation and optimizing the process. It should mention that our activation function is RELU function. Within the backpropaga-

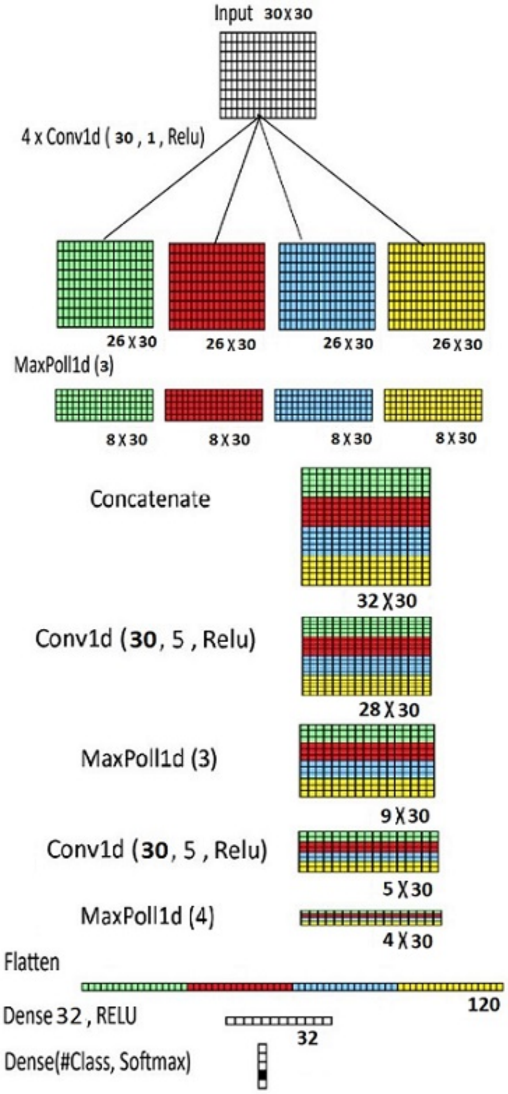

Figure 6: The Architecture of CNN layers

tion step of the algorithm, we defined a well-known loss function called softmax-cross-entropy in addition to a regularization term for weight decay with considering the norm of weights. Before minimizing the loss function, we used AdamOptimizer to train our optimizer for handling learning rate more efficiently than other similar methods such as Gradient Descent. Moreover, to train our dataset, it has been split to train and test parts by using a function in Scikit learn library. For the Reason That we did not have any hyperparameter to regularize, we did not need to have a validation set. Therefore, we just split our dataset into two parts of train and test.

4.2 Other Methods

One of the common ways to check the quality of a model is by comparing it to other existing models. In this case, we compared the accuracy of CNN prediction to SVM and KNN. Furthermore, after defining tf-idf as a document representation, SVM, KNN, and DT were implemented again. Thus, this comparison is shown in Table 1 that indicates the accuracy of them. According to the outputs, CNN works better than other methods, because the accuracy of this model is 0.69 which equals with SVM and KNN but more than these in tf-idf approach.

4.3 Results Comparison

After changing thresholds, include train-test share, batch size, or epoch size, as well as examining both approaches of representation (word2vec and tf-idf), we received results shown in Table 1. The outcome indicates that CNN's accuracy for predicting ratings founded on reviews is a bit higher than those of traditional models. This upshot (%69 of accuracy) has been achieved by the number of 10 for both batch and epoch sizes and %70 shares for training data.

Model	Accuracy
CNN	**0.69**
SVM	0.69
KNN	0.69
KNN (tf-idf)	0.66
DT (tf-idf)	0.64

Table 1: Experimental Results

5 DISCUSSION AND OUTLOOK

Due to the instant growth changing customers' behavior from buying in person to shopping online, the importance of opinion mining of online users is increasing for all suppliers, retailers, and consumers these days. The idea of our work is predicting rating from a text that has a diverse application like scoring spam detection, scoring fraud detection, and many others else. In this study, we have tried to find an answer for the main question of whether neural network algorithms specifically CNN can perform more efficiently than other widespread machine learning methods such as SVM, KNN, and DT in text mining. Therefore, we have compared the accuracy of these models. For executing this comparison task, we have applied two distinct representations (word2vec and tf-idf). In summary, the accuracy results can be seen in Table 1 shows that CNN has a more accurate performance than other methods. The most remarkable limitation of this study was the relatively small dataset. Since our model was a deep learning algorithm that needs a huge amount of data, the accuracy was impacted by lacking this amount. For future works, we would like to work on other Neural Network models, specifically Recurrent ones like LSTM, that figured out an alternative model while we were exploring the papers.

REFERENCES

[1] Wanliang Tan, Xinyu Wang, Xinyu Xu, *Sentiment Analysis for Amazon Reviews*, 2018, 5pages.

[2] Xing Fang, Justin Zhan, *Sentiment analysis using product review data*, 2015, 14 pages

[3] Marinho W, Martí L, Sanchez-Pi N. *A Compact Encoding for Efficient Character-level Deep Text Classification* Proc Int Jt Conf Neural Networks, 2018, 9 pages.

[4] 1. Rozi MF, Mukhlash I, Soetrisno, Kimura M. *Opinion mining on book review using CNN-L2-SVM algorithm* J Phys Conf Ser. 2018, 9 pages.

[5] Yang Z, Zhang Z-K, Zhou T. *Anchoring bias in online voting* 2012, 100 pages.

[6] https://datafiniti.co/.