Paper:

Saumya, S., Singh, J. & Dwivedi, Y. (2019). Predicting the helpfulness score of online reviews using convolutional neural network. *Soft Computing*

http://dx.doi.org/10.1007/s00500-019-03851-5

Predicting the helpfulness score of online reviews using convolutional neural network

Sunil Saumya Jyoti Prakash Singh Yogesh K. Dwivedi

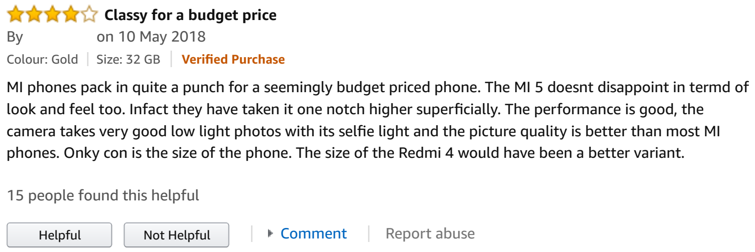
February 2019

Abstract

The smart cities aim to provide an infrastructure to their citizen that reduces both their time and effort. An example of such an available infrastructure is electronic shopping. Electronic shopping has become the hotbeds of many customers as it is easier to judge the quality of the product based on the review information. The purpose of this study is to predict the best helpful online product review, out of the several thousand reviews available for the product using review representation learning. The prediction is done using a two-layered convolutional neural network model (2-CNN). The review texts are embedded into low-dimensional vectors using a pre-trained model. To learn the best features of the review text, 3 filters are used to learn tri-gram, four-gram, and five-gram features of the text. The proposed approach is found to be better than existing machine learning based models which used handcrafted features. The very low value of mean squared error (MSE) confirms the prediction accuracy of the proposed method. The proposed method can be easily applied to any kind of review as the features are calculated only from the review text and not from other domain knowledge. The proposed model helps in predicting the helpfulness score of new reviews as soon it gets posted on the product review page.

1 Introduction

With the advent of the internet and technology, there is a process of making our cities and villages “smart". Here, the term “smart" means providing better living conditions to the citizens. The people of smart cities or villages make their decision about any product or services based on their quality. That means they find how the peer customers have evaluated the product or service and based on that make their own decision. One way of finding the customers feedback is “online consumer reviews". There are some other ways also to get feedback of product or services such as one-to-one interaction, news channel, or advertisements. However, according to a survey by BrightLocal (BrightLocal, 2016), 88% of consumers refer online reviews for decision making. A review aggregation website (like *yelp.com*, *mouthshut.com*) is one that maintains online reviews (or opinions) about the products and services such as mobiles, books, hardware, software and so on (Kizgin et al., 2018; Lee and Choeh, 2014). Apart from this, E-commerce websites also maintain online reviews of those products which they are selling. Some well-known E-commerce websites are *amazon.in (Amazon)*, *snap-deal.com (Snapdeal)*,



★★★★☆ **Classy for a budget price**

By on 10 May 2018

Colour: Gold | Size: 32 GB | Verified Purchase

MI phones pack in quite a punch for a seemingly budget priced phone. The MI 5 doesnt disappoint in termd of look and feel too. Infact they have taken it one notch higher superficially. The performance is good, the camera takes very good low light photos with its selfie light and the picture quality is better than most MI phones. Onky con is the size of the phone. The size of the Redmi 4 would have been a better variant.

15 people found this helpful

Figure 1: A sample review of Amazon

flipkart.com (Flipkart), and so on. These websites store online consumer reviews for various purposes, for example: (i) it creates database of customer's purchase history and use that in recommendations (Saini et al., 2017) or, (ii) it may act as a major information source for customers to help them in making their purchase decision (Chua and Banerjee, 2016; Krishnamoorthy, 2015; Shareef et al., 2018; Singh et al., 2017) and so on. In online shopping, one often checks customer's experience, in the form of reviews, before deciding which product to buy (BrightLocal, 2016). However, the quality of online reviews is very inconsistent. It ranges from superior elaborated assessments (or helpful reviews) to continual statements of product specification (or non-helpful reviews) or, in the worst case, it can be a spam (Lee and Choeh, 2014; Saumya and Singh, 2018) to promote the unpopular product or defame a good quality product.
In many cases, the number of reviews received by a product is very large. For example, as we can see in Table 1, a product *Amazon Fire TV stick* have received 144,805 reviews, *Cards against humanity* have received 37,635 reviews and similarly, *Redmi 4 (32GB)* have also received more than 50,000 reviews. That means a huge amount of information is available online through which customers can judge the quality of products and make their purchase decision. However, (i) reading all such reviews present for a product to make the purchase decision is nearly impossible and, (ii) due to these overwhelming range of reviews, the most helpful reviews may be buried in ample amount of non-helpful reviews which prevent the effective use of reviews. To mitigate these issues, most review websites started prioritizing the submitted reviews based on users' evaluations. For example, Amazon in 2007, came with helpfulness vote mechanism in which reviews of products are being sorted based on their received votes. They keep a simple question at the end of each review "was this review helpful to you?" and give two buttons *Helpful* or *Not Helpful*. A reader can give the votes in favor or against of a review by clicking any of the given buttons. For example, a sample review is shown in Fig. 1 which received 15 helpful votes. Based on the received votes the reviews are being sorted and hence, the most helpful reviews would be bubbled at the top in the review list. This mechanism increased the Amazon's economy by $2.7 billion per year (Spool, 2009). Later, other websites like *Flipkart* and *Snapdeal* also started allowing readers of a review to inform whether they believe the review is helpful or not by voting for it or against it.

However, helpfulness of online reviews cannot be considered as a golden spoon, which guarantees to increase readership of reviews and hence, attract more votes (Singh et al., 2017). This is because customers are selective in deciding what to leave out and what to read. According to BrightLocal survey (BrightLocal, 2016), people read less than 10 reviews before making their purchase decision. But, in practice, popular products

Table 1: Number of reviews received by Amazon best seller products

Product Category	**Product Name**	**Number of Reviews**
Electronic	Amazon fire TV stick	144,805
Toys & Games	Cards against humanity	37,635
Video Games	$20 play station store	23,390
Books	1984 (Signet Classics) by George Orwell	6,254
Mobiles & Accessories	Redmi 4 (32GB)	54,400

receive reviews in thousands but, all reviews are not good enough to attract readership. Moreover, the current review system is biased by the posting time of reviews. That means reviews which are posted earlier get more attention by the readers whereas, late posted reviews are ignored by appending them at last in the review list. This phenomenon is termed as Matthew effect (Wan, 2015). By analyzing the dataset of Amazon (2187 reviews), (Wan, 2015) identified that early posted lengthy reviews received an unreasonably higher percentage of votes due to Matthew effect and once found as most helpful, could maintain their top positions throughout the life cycle of products due to the Ratchet effect (Freixas et al., 1985). This prevents buyers to get insights from later posted helpful reviews, as a result of that the original purpose of the helpfulness voting is violated. To mitigate the negative impacts of these effects, there is a strong need of a mechanism that places the best k helpful reviews at the front of the review list based on their quality, and not by their helpfulness votes.

A number of researches have been proposed for predicting the most helpful reviews. Most of them were mainly focused on extracting most relevant features upon which prediction was calculated from various determinants like textual contents, reviewer characteristics and review characteristics (Cao et al., 2011; Chen and Huang, 2013; Chua and Banerjee, 2015; Korfiatis et al., 2012; Singh et al., 2017). Saumya et al. (2018) proposed to use product description features and question-answer features along with earlier features. The previous works mainly used statistical methods such as support vector machine for regression or least square regression (Cao et al., 2011; Ghose and Ipeirotis, 2011) or conventional supervised machine learning methods such as random forest, naive Bayes', or gradient boosting (Allahbakhsh et al., 2015; Baek et al., 2015; Chua and Banerjee, 2016; Korfiatis et al., 2012; Krishnamoorthy, 2015; Ullah et al., 2015) for helpfulness prediction. As per our knowledge, there is only one reported work which has used the artificial neural network for the same purpose (Lee and Choeh, 2014). They used multilayer perceptron networks for helpfulness prediction.

The earlier approaches suffer from two main inadequacies. First, from the feature extraction point of view, the features used for prediction were mostly hand-crafted hence, it is bound to be biased and costly. For example, features can be a number of words in the review, readability of review, polarity and so on. The performance of most of the Machine Learning methods depends on how accurately these features are identified and extracted. Second, from the semantic point of view, none of the earlier works focused on the contextual analysis of review text, instead, they focused only on content-based factors from review text.

This paper intends to address the aforementioned inadequacies inherent in the existing system by learning continuous sentence representation using a convolutional neural network (CNN) based model. The other way of learning continuous sentence representation of reviews is by using the Recurrent neural network (RNN) (Mikolov et al., 2010; Ren and Ji, 2017). A review generally describes features of the products pointwise. For example, a mobile product review first may describe the camera feature then its battery and so on. The points (or sentences) written in reviews do not have the long dependency. However, RNN, specially LSTM, is known for learning long sequences of input data. Hence, we preferred to use CNN. The current model also considers those cases where the sentences in a review are dependent. But, as we fixed the maximum length of review to be 50 (average review length), which is not very long, the CNN performs better compared to other (Kumar and Singh, 2018). The possible advantages of using CNN for helpfulness prediction of reviews are three folds. First, CNN uses the number of hidden layers for automatic feature extraction, which can preserve the complex global semantic information that is hard to obtain from conventional hand-crafted features. Second, inputs to the CNN have distributed word vectors (or word embeddings), which can be trained on a significant large corpus of raw text, thus reducing the need for labeled data to some extent. Third, CNN can learn continuous sentence representation, leveraging discourse models simultaneously.

This paper shows that substantial enhancements can be attained by learning continuous sentence representation using a CNN model. Based on the predicted helpfulness value, we construct the new review list and prioritize the review accordingly. It gives a fair chance to each review to be visible at the front page of reviews. The current system is evaluated with the datasets of two popular e-commerce websites *Amazon* and *Snapdeal*.

The experiments show that the proposed CNN model outperforms several state-of-art models, substantiating the edge of deep neural models in preserving semantic features.

The rest of the article is structured as follows: Section 2 explains the related work of review helpfulness prediction. Section 3 presents the explanation of our proposed CNN model. Section 4 introduces the tools used for implementations and then reports various experimentation results by tuning several hyper-parameters. Section 5 discusses the main findings and implications of those. Finally, in Section 6 we conclude this work by pointing few limitations and further research scopes.

2 Related Work

In online shopping, customers generally look for two factors before making a purchase decision of a product. The first is a specification of the product, which must satisfy customer requirements and other is the opinion of other customers who have bought that product before. These opinions are also termed as online reviews. In this work, we are predicting the best reviews out of thousands of reviews available for a product. In this section, we discuss some of the important works and give an intuition behind doing this study. We have categorized this section into two parts. The first part talks about the most relevant works and the second part talks about the need for a neural network for predicting the best helpful reviews.

2.1 Best helpful reviews prediction

A number of research works have been reported which examine the factors which influence the review to become most helpful (Ghose and Ipeirotis, 2006; Kaushik et al., 2018; Saumya et al., 2016; Wu, 2017). For example, (Ghose and Ipeirotis, 2006) showed the impact of review subjectivity on review helpfulness. They proposed two ranking mechanisms, customer-oriented ranking mechanism, and manufacturer-oriented ranking mechanism, for ranking online reviews. In the first approach, the reviews were ranked based on their expected helpful votes and in later, the reviews were being ranked based on their effect on product sales. They used Random-forest based classifier for their experiments. They found that more informative reviews have mixed subjective and objective elements. Similarly, Liu et al. (2007) explored review informativeness, readability, and subjectivity for categorizing reviews into low and high qualities. Some other factors like review lengths, timeliness of review, reviewer expertise and reviewer's writing style were also considered for predicting most helpful reviews (Liu et al., 2008; Otterbacher, 2009; Siering et al., 2018).

Forman et al. (2008) found that identity disclosure of reviewer attracts other community members who rate the products more positively. A report said that the product's average rating and review helpfulness are consistent with each other (Danescu-Niculescu-Mizil et al., 2009). On the paradigm of product types (search and experience[1]), (Mudambi and Schuff, 2010) used statistical approach and found that the review extremity and review length have a collaborative effect on review helpfulness. For search products, the review length and extreme ratings have a positive impact on helpfulness as compared to experienced products. So, they concluded that along with review length and review extremity, the helpfulness was also dependent on product types. Ghose and Ipeirotis (2011) analyzed linguistics features of review to find the factors of review helpfulness. They found that medium length reviews with a few spelling mistakes are more influential for new customers compare to very long and very short reviews with more spelling mistakes. However, (Korfiatis et al., 2012) analyzed that review readability was more influential than its length for review helpfulness.

In most of the researches the sentiments of the review was investigated for its effect on product sales and review helpfulness (Cao et al., 2011; Li et al., 2013; Schumaker et al., 2012). Cao et al. (2011) used text mining techniques and found that extreme opinions are more influential for helpfulness than mixed or neutral opinions. Liu et al. (2013) found a research gap that most of the works mine sentiments from a collection of reviews posted during specific periods only. They did not analyze the change in sentiments when the collection of reviews evolve. To overcome this, they developed an adaptive sentiment analysis models. Their results show that the proposed adaptive method can capture sentiment variations from newly posted reviews, which helps in predicting the product sales and services more accurately. Siering and Muntermann (2013) extracted the sentiment of reviewer regarding the quality of products using Tobit regression. They found that reviews with the discussion about the products quality received more helpful votes.

Other than content-based factors, (Li et al., 2013) analyzed the source based factors for review helpfulness prediction. They found that less content and high comprehensible reviews were more helpful. Wang et al. (2013) proposed a web-based system for

[1] A search product is one whose quality and features can be evaluated before purchase. In contrast, for experience products evaluation can be done after purchase.

automatically extracting most relevant customer opinions using several features. They named it as “SumView". In line with that (Wan et al., 2018) discussed the way online reviews communicate with each other and how reviewers opinions evolve over time. Wu et al. (2013) modeled a two-stage mechanism to investigate the importance of online reviews in consumer's willingness to pay. Hu et al. (2014) developed a model using multiple equations to establish the relation between sentiments, ratings and product sales. They found that ratings did not directly affect the sales. Another important observation was the most recent and most helpful reviews highly influence the product sales. A recent work proposed by Kaushik et al. (2018) studied the impact of a sequence of helpful reviews on product sale. Lee and Shin (2014) conducted a survey to understand why readers always accept quality-based review. They used a statistical approach to establish the relationship of review quality, review sentiment with review helpfulness. They found that high-quality positive reviews increase the product sale in comparison to low-quality ones. Similarly, Tsao (2014) found that the negative consumer reviews on movie selection are stronger than that of positive consumer reviews. Wan and Nakayama (2014) explored the reliability of review rating. They found that ratings of most helpful reviews are biased and significantly higher than the genuine. They also found that ratings of most helpful favorable reviews are biased by first few reviews due to the common human tendency to believe excessively on the first piece of information offered when making decisions. The similar effect was found for most helpful critical reviews also as if first few customers have rated it negative others start following.

Krishnamoorthy (2015) examined the linguistic features like an adjective, action verb, state verb and then accumulated them to built a predictive model for helpfulness prediction. Some other factors used by them were review extremity, review age, readability measures, and subjectivity. In addition to review characteristics, (Guo and Zhou, 2017; Huang et al., 2015; Roy et al., 2018) used reviewer characteristics to capture their joint effect on helpfulness. Huang et al. (2015) used Tobit regression as a predictive model. They found that review length up to a certain threshold has a significant effect on review helpfulness. However, this effect diminishes beyond a given threshold. Similarly, (Guo and Zhou, 2017) found that the impact of review length and review valance on review helpfulness was positively moderated by linguistic style similarity, while expertise similarity negatively moderated the effect of review valence and review length on review helpfulness. Allahbakhsh et al. (2015) introduced a framework which has three main components. First, a robust rating score calculation. Second, the reviewer's behavior analysis and third, reviewer trust rank. Weathers et al. (2015) explained two terms diagnosticity and credibility. Diagnosticity defines uncertainty and credibility defines the reviewer's expertise and trust. They found that these two terms are often ineffective for review helpfulness. Chua and Banerjee (2015) investigated three factors review length, review rating, and reviewer reputation. Using multiple regression and Tobit estimation, they found that reviewer reputation and review length have a positive effect on helpfulness whereas, review rating was negatively associated. Qazi et al. (2016) considered both a qualitative and quantitative approach while building a regression model for helpfulness prediction. They used Tobit regression to categories the reviews into three parts: regular, comparative and suggestive. A regular review is simply an opinion, a comparative review explains the differences between two products, and a suggestive review is a suggestion to buy or not to buy. They found that all three types of reviews have their own significant effect on the purchase decision. Chua and Banerjee (2016) used statistical based approach and investigated three factors, review sentiments, product type, and information quality and their effect on review helpfulness. They found that review helpfulness varies as a function of review sentiments and it was independent of product types.

Singh et al. (2017) predicted the helpfulness of reviews using several textual features. They also explored the Mathew effect (Merton et al., 1968) in the current review listing system (Chua and Banerjee, 2017). They used two regression techniques linear regression and gradient boosting regression. They found that polarity, subjectivity, readability, entropy, and average rating are the most influential features for helpfulness. Zhang and Lin (2018) proposed a multilingual approach for predicting helpfulness of reviews using statistical methods. Similarly, Liu et al. (2018) proposed multi-view ensemble learning for product defect identification. They used several classifiers like Naive Bayes, Random Forest, Logistic Regression, Support Vector Machine and k-Nearest-Neighbors for their experiments.

2.2 Neural networks for review representation

Nowadays, Neural networks are extensively being used in a number of natural languages processing tasks (Chen et al., 2018; Li et al., 2017; Ren and Ji, 2017). Text representation learning using neural networks have been proven effective and have replaced conventional task-specific feature engineering (Prieto et al., 2016). In contrast to feature engineering, representation learning does not need domain expertise. In representation learning, a continuous real-valued vector can be incorporated as a feature to learn a continuous representation of word, sentence, and documents. As per our knowledge, there is no reported work for review helpfulness prediction using review representation learning. All existing works have used a statistical approach or conventional machine learning approach, except the one proposed by (Lee and Choeh, 2014) who used multilayer perceptron model, for helpfulness prediction. Lee and Choeh (2014) used manually crafted features from review data and its metadata as an input to their proposed model. But, they did not use any representation learning as such.

To the best of our knowledge, this is the first study which uses continuous review representation for review helpfulness prediction using a convolutional neural network (CNN) model. The proposed CNN model (Kim, 2014) can preserve the complex global semantic information and eliminates the need for complex hand-crafted features. CNN automatically captures the n-gram information present in a review.

3 Research Methodology

The proposed convolutional neural network model learns a continuous representation of review, which is used as features to predict the helpfulness value of each review. As shown in Fig. 2, the proposed model mainly contains two parts. The first part takes word vector as an input and produces the corresponding sentence vectors (Section 3.1) and the second part takes sentence features of each review as an input to CNN and predicts corresponding helpfulness value (Section 3.2).

To evaluate the proposed model, we used the dataset of the paper (Saumya et al.,

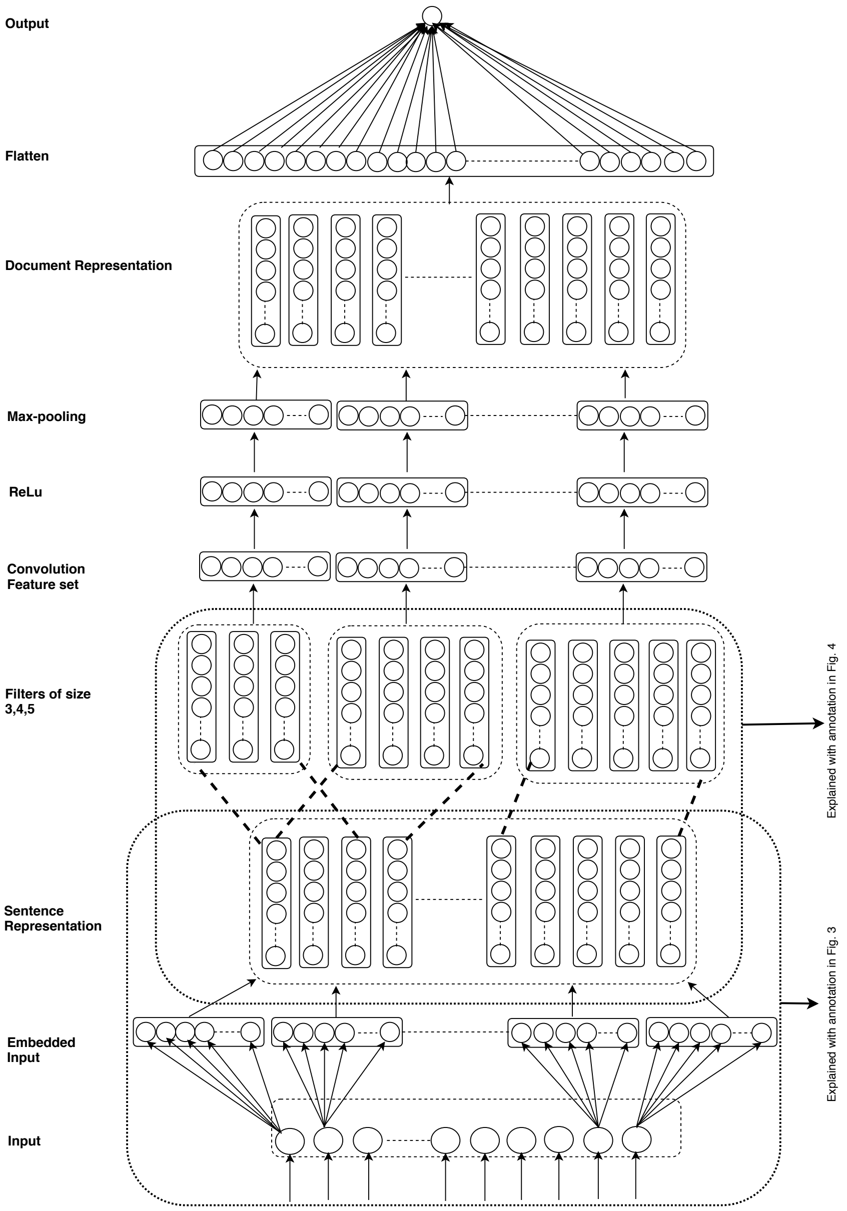


Figure 2: Flow diagram of proposed system

Table 2: Number of reviews collected for each products

Products	Number of Reviews	
	Amazon	Snapdeal
Power bank	3371	1370
Mobile phone	5031	2031
Memory card	12271	8951
Baby product	1301	139
Book	5441	163

2018) which were collected from two popular Indian e-commerce websites *Amazon* and *Snapdeal*. The dataset contained the following fields: product id, customer id, review text, number of votes received by the review, number of comments received by the review, and the rating given by the reviewer. However, we used only two fields for experimentations that are review text and the number of helpful votes received by each review. We ignored the number of comments and the content of comments in order to predict the helpfulness of review because to receive comments a review must be visible to the readers for the considerable time. But, the current research aims to prioritize new reviews as soon it appears on the product review page. The text of the review is the input to the proposed model and the output is the number of helpful votes that the review has received at that point (Gao et al., 2017). The problem is in some sense, a regression problem. The underlying assumption is that helpful reviews would have more votes and not-helpful or spam reviews would have fewer votes. As our target was to predict best k helpful reviews and rank them accordingly, we believe the number of votes is the correct quantity to predict (Gao et al., 2017; Ghose and Ipeirotis, 2011; Mudambi and Schuff, 2010). The rest information was removed from the dataset in data pre-processing steps. We also removed all unicode characters and images present in the dataset using a Python program. Therefore, the user's privacy (user's id or user's profile info) has not been used anywhere. In total, there were 29,215 reviews of *Amazon* and 12,886 reviews of *Snapdeal* for five different products mobile phones, baby products, power bank, memory card, and books. The data dimension used for this study is shown in Table 2. Most of the reviews received votes in the range of 0-10 whereas, some reviews received uncommonly high votes (like 500 votes or 1000 votes). To ensure the smooth learning of proposed model, we normalized the uncommonly high-vote reviews to triple the non-zero average of helpful reviews (as it was done by (Saumya et al., 2018)).

3.1 Word to sentence representation

We used word embeddings of the review text as input to the system (Bengio et al., 2003; Levy and Goldberg, 2014). Word embeddings are a representation of words into vectors which are of low dimensional, real-valued and continuous. For word vector representation, we first created bag-of-words from all unique words present in our review texts. Then for each word x, we created a look-up matrix M to obtain its embedding $e(x) \epsilon R^D$, where $M \epsilon R^{D \times S}$ is embedding parameter, S is the total words in vocabulary and D is the dimension in which each word is represented. The look-up matrix M can be initialized in two ways. First, M can be randomly initialized from uniform distribution (Socher et al., 2013) in a typical neural network model and it generally initialized as

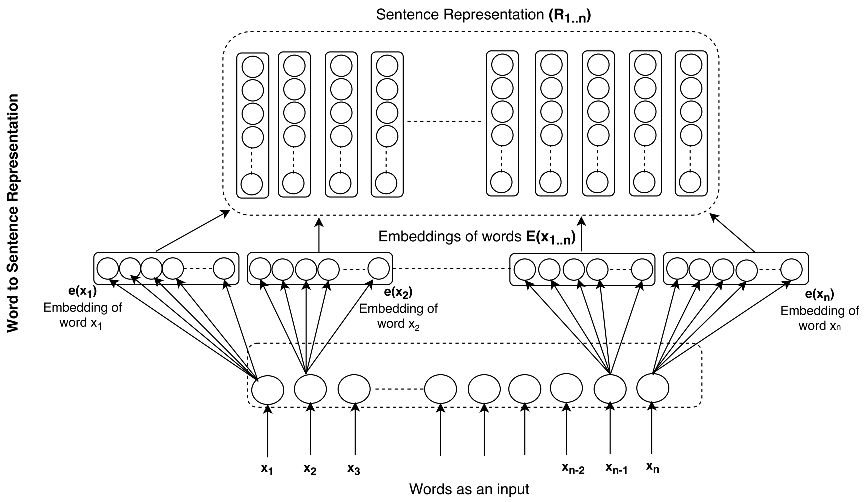

Sentence Representation ($R_{1..n}$)
Word to Sentence Representation
Embeddings of words $E(x_{1..n})$
$e(x_1)$ Embedding of word x_1
$e(x_2)$ Embedding of word x_2
$e(x_n)$ Embedding of word x_n
x_1
x_2
x_3
x_{n-2}
x_{n-1}
x_n
Words as an input

Figure 3: Converting words into sentence representation of review (part of Fig. 2)

a weight between inputs to first hidden layer. Second, M can be a pre-trained matrix from a large corpus with embedding learning algorithms (Mikolov et al., 2013). In our case, we have used pre-trained look-up matrix M for experiments. A detailed view can be seen from Fig. 2. Here, each input has been represented in a number of neurons in the *Embedded Input*. The pre-trained matrix M has been used as weight between *Input* layer and *Embedded Input* layer.
We used two different embedding algorithms to get pre-trained look-up matrix M, one is Word2Vec (Mikolov et al., 2013) and another is GloVe (Pennington et al., 2014). To implement Word2Vec we used gensim package available in python and trained the model with 26807 words (total tokens in our corpus) and represented it into the 100-dimensional vector. We did not implement GloVe embeddings on our own instead, we used pre-trained GloVe embeddings *"glove.6B.100d.txt"* [2]. It is trained by *Google* on 6 billion Wikipedia words and each word has been represented into 100-dimensional vectors. The popularity of GloVe is increasing as a pre-trained word vector input to any deep neural model (Kumar et al., 2016; Bowman et al., 2015). We performed our preliminary experiments on both Word2Vec as well as GloVe and used them as a weight between the input layer and first hidden layer. However, from experimental results, we found that our model performed better on GloVe embedding than Word2Vec (See Table 3). Therefore, we preferred GloVe over Word2Vec for further experiments. Moreover, using GloVe also reduces our computation overhead due to its free availability. The detailed experimentation results have been discussed in the Result section (Section 4).

To represent complete review into matrix form we concatenated the embeddings of each word present in a review one after other (Fig. 3). For example, suppose a review R has n words as $x_1, x_2, x_3, ..., x_n$. The embeddings of each words has been represented as follows:

$$E(x_{1..n}) = e(x_1), e(x_2), e(x_3), ..., e(x_n) \quad (1)$$

Where, $E(x_{1..n})$ represents the embedding of all words present in a review and $e(x_1), e(x_2), e(x_3), ..., e(x_n$ represents the embedding of individual word. So, after concatenating all embedded

[2] It is freely available on *nlp.stanford.edu/projects/glove/*

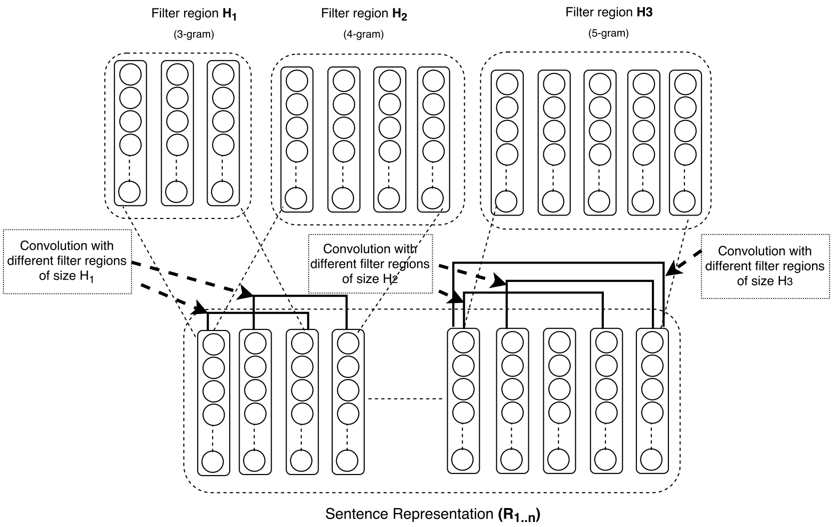

Filter region H1
(3-gram)
Filter region H2
(4-gram)
Filter region H3
(5-gram)
Convolution with different filter regions of size H1
Convolution with different filter regions of size H2
Convolution with different filter regions of size H3
Sentence Representation (R1..n)

Figure 4: Convolution with different size of filters H_1, H_2, and H_3 (part of Fig. 2)

words $E(x_{1..n})$ a sentence representation of review r can be presented as:

$$R_{1..n} = e(x_1) \oplus e(x_2) \oplus e(x_3) \oplus ... \oplus e(x_n) \tag{2}$$

The transformed matrix $R_{1..n} \epsilon R^{D \times W}$ is the sentence representation of a review having n-words where W are the total words in review $R_{1..n}$ and D is the dimension of each word. The pictorial view can be seen from Fig. 3, where embedded inputs are concatenated one after other and formed sentence representation of review. Then this sentence representation we used as an input for the CNN model which give a predicted score for each review.

3.2 Sentence features to helpfulness prediction

We used the CNN model for preserving semantic features of sentence (Kalchbrenner et al., 2014; Socher et al., 2013). The convolution process on sentence are normally used to conserve n-gram information (Collobert et al., 2011; dos Santos and Gatti, 2014; Ren and Ji, 2017). Extracting n-gram information has very wide application in NLP tasks. Hence, we also used them to extract high dimensional local features from a continuous sentence matrix. The proposed model uses two layers of convolution (say 2-CNN). The first convolution layer is used to convert continuous sentence representation into document representation and then the second convolution works on that document representation matrix to calculate the helpfulness score for each review. We used three convolutional filters of region sizes H_1, H_2, and H_3 as shown in Fig. 4.

A convolutional filter F act as a linear layer and is represented by a weight matrix with which we convolve the concatenated sentence representation of review $R_{1..n}$. Because, texts have a one-dimensional structure where words sequence matter, we kept the size of the filter F in the form of, $F \ \epsilon \ R^{H \times D}$. Where, H is the region size (H_1, H_2, and H_3) which refers to number of rows (or representing word) of created sentence matrix $R_{1..n}$ that would be filtered and D is the dimension of each row which is same as word embedding $E(x_{1..n})$ dimension. In our case we took region size 3, 4 and 5 for H_1, H_2, and H_3 respectively (Fig. 4). Now, suppose for the region size H, filter F starts convolving on created sentence matrix $R_{1..n}$. The convolution occur for all possible window of sentence matrix $R_{1..n}$ in top to down manner. The output of this convolution is, convolution feature map (T) (or simply feature maps), calculated as follows:

All possible windows of sentence matrix $R_{1..n}$ of region size H:

$$H_1 = (R_{1..H_1}, R_{2..H_1+1}, R_{3..H_1+2}, ..., R_{n-H_1+1..n}) \tag{3}$$

Similarly, for region size H_2 and H_3 the possible window size can be following:

$$H_2 = (R_{1..H_2}, R_{2..H_2+1}, R_{3..H_2+2}, ..., R_{n-H_2+1..n}) \tag{4}$$

$$H_3 = (R_{1..H_3}, R_{2..H_3+1}, R_{3..H_3+2}, ..., R_{n-H_3+1..n}) \tag{5}$$

The feature maps formed from filter F of region size H_1:

$$T_1 = R_{1..n} \cdot H_1 \tag{6}$$

Where, T_1 is the first feature map formed from filter F of region size H_1. First, the three-word filter H_1 overlays across the first three rows of sentence matrix $R_{1..n}$. Next, it performed the element wise multiplication for all elements and sum them up to obtain one number $T_{1,1}$. This number $T_{1,1}$ is the first feature in the feature maps T_1. Next, the filter moved down one 1 row and overlays across the next three rows of sentence matrix $R_{1..n}$ and formed second feature $T_{1,2}$ of feature map T_1. Similarly, we convolved other features of feature map T_1. In general the total number of features in a feature map T_1 is calculated as:

$$T_1 = n - H_1 + 1 \tag{7}$$

Where, n is the total words in sentence matrix $R_{1..n}$ and H_1 is the region size of the filter. The size of the created feature map T_1 is given as $(n - H_1 + 1) \times 1$. That means in each feature map T there is $(n - H_1 + 1)$ rows and 1 column.

To obtain non-linearity, an activation function *ReLu* (Nair and Hinton, 2010) was applied on feature map T_1 to obtain the output A_1.

$$A_1 = ReLu(T_1) \tag{8}$$

ReLu activation function $f(x)$ is defined as: $f(x) = max(0, x)$. Where, for negative value of x the function returns 0 and for positive value of x the function returns x itself. The shape of A_1 was same as of T_1 as $(n - H_1 + 1)$. That means the dimensionality of A_1 was dependent on total words in n and filter region size H_1. In other words, the dimensionality of A_1 was varying for reviews of different lengths and filters of different region sizes. Hence, to address this problem we applied *1-max pooling* function (Graham, 2014) on each output obtained from *ReLu* function. The output O_1 of *max-pooling* function extracts maximum value from each window Z of matrix A_1.

$$O_1 = max(A_1, Z) \tag{9}$$

Similarly, we obtained other two outputs O_2 and O_3 for filter regions H_2 and H_3 respectively. Then, we concatenated the outputs O_1, O_2 and O_3 which results in another matrix which we termed as document representation of review.

The second layer of convolution is applied to the obtained document representation matrix. This time we used only one filter region of size 3. All the outputs obtained at second *max-pooling* layer were then flattened into one single array as shown in Fig. 2. Then the inputs of *Flatten* layer was projected to three *Dense* (or fully connected) layers. After each *Dense* layer we used *Dropout* layer (Srivastava et al., 2014) except the last one. Dropout is a regularization technique in which during each iteration, we drop a set of neurons selected at random. By dropping we simply mean as they do not exist. Dropout prevents over-fitting and speeds up learning by randomly dropping some neurons with the probability p in the training phase. For each layer in the neural network, the value p may be different. This prevents conversing the weights to same positions, as for every learning example a dissimilar set of neurons is dropped at random, which results in a strong set of features that can generalize with new unseen data in the better way. We kept only one neuron at last *Dense* layer as our target was to predict one score for each review. We did not apply any activation function at *Output* later and collected the final score for each review.

3.3 Loss Function and Optimizer

After the network (or model) creation, our next objective was to compile the model. Generally, to compile any model we need two parameters, one is loss function and other is optimizer. We have explained each below.

A loss function explains how we are penalizing our output. In other words, a loss function is used to check how close or far the results of our model is from the actual results. We then back-propagated (McCollum, 1997) the calculated loss at the output layer through our network, to adjust its weights and make it get closer to the actual output the next epoch around. The weight adjustment was based on the *Adam* optimizer (Kingma and Ba, 2014). Due to the regression nature of the problem, in our case, we used *Mean Squared Error* (MSE) (Wang and Bovik, 2009) loss function between the expected output of proposed 2-CNN network and the calculated output. While training 2-CNN, our main objective was to minimize the MSE over a batch of training data $(r_i, y_i)\ \forall\ i = 1\ to\ n.$ where, r_i, represents the review, was the input the 2-CNN and y_i, represents the obtained helpful votes of review, was the corresponding target.

$$Mean\ Squared\ Error\ \ : MSE(E) = \frac{1}{n}\sum_{i=1}^{n}(f(r_i) - y_i)^2 \tag{10}$$

We used *Adam* optimizer (Kingma and Ba, 2014) to back-propagate the calculated loss E. For every batch of training data, the parameter were updated by *Adam* update rule which is given as follows:

$$\theta^{t+1} = \theta^{t} - \frac{\eta}{\sqrt{\hat{v}_w} + \epsilon}\hat{m}_w \tag{11}$$

Given,

$$\hat{m}_w = \frac{m_w^{t+1}}{1 - \beta_1^t} \tag{12}$$

and,

$$\hat{v}_w = \frac{m_v^{t+1}}{1 - \beta_2^t} \tag{13}$$

Where:

ϵ : is small constant to avoid division by 0
β_1 : is forgetting factors for gradients
β_2 : is second moments of gradients
η : is the learning rate.
θ : represents the parameters.

We used the default value of ϵ as 10^{-8}, β_1 as 0.9, and β_2 as 0.999.

4 Results

We performed experiments to estimate our review representation learning model by applying it to find best helpful reviews. First, we conducted two separate experiments for all data of *Amazon* and *Snapdeal*. Later, we conducted individual experiments for five different products of *Amazon* and *Snapdeal*. We made an analysis of proposed 2-CNN based neural model and one layer of the convolutional model (say 1-CNN) to justify why we use 2-CNN in our case. We also made an analysis of our approach and conventional gradient boosting regression.

4.1 Experimental setup

We used *python 2.7*, a high-level programming language, to implement our proposed 2-CNN model. In particular, we used *Keras* library to create and fit our model on reviews of *Amazon* and *Snapdeal*. *Keras* is a high-level neural networks API which uses *Tensorflow* and *theano* as a backend. For our case, we used *Tensorflow*. In our dataset, the length of each review was different. But, the CNN model we built in Keras accepts only fixed lengths inputs. To tackle this problem, we fixed the maximum review length as 50. That means reviews which have less than 50 words are padded with zeros to make them 50 lengths. Similarly, reviews which have more than 50 words, only the first 50 words are kept curtailing the rest. Apart from this, we used *Word2Vec* method of *Gensim* package to create the vector representation of words. The created embeddings from *Gensim* and publicly available embeddings from *GloVe* were used as inputs to 2-CNN.

4.2 Model Experiments

The main intuition behind any neural model is to iteratively learn weights and stop once a threshold is received. This threshold can be achieved in various ways based on the nature of the problem. For example, in some problems, we say threshold received once the system converges, or in another case, it can be the number of iterations (or epochs) and so on. In our case, we iterated our model for different epochs (say 100, 200, 300, 400, 500) and batch sizes (say 50, 64, 128). The best results we got for 100 epochs with batch size 128. Hence, results discussed in this section are for the settings

Table 3: Convolutional and Dropout layers effect on helpfulness prediction of Amazon and Snapdeal reviews

Approach	Embedding	MSE	
		Amazon	Snapdeal
1-CNN	Gensim	1.562	2.258
1-CNN	GloVe	0.248	0.386
1-CNN + Dropouts	Gensim	1.527	2.226
1-CNN + Dropouts	GloVe	0.239	0.424
2-CNN	Gensim	1.126	2.043
2-CNN	GloVe	0.216	0.273
2-CNN + Dropouts	Gensim	1.099	2.114
2-CNN + Dropouts	GloVe	**0.213**	**0.223**

100 epochs and 128 batch size. We stored the MSE value as a system accuracy. We used 10-fold cross-validation to minimize the biasness (Kohavi et al., 1995). The system was trained with 9 batches of 10-fold cross-validation and validated with remaining one batch. We performed this for all permutations. The results shown here are the MSEs for the test set. In Table 3, we present the performance of system on *Amazon* and *Snapdeal* datasets. We used pre-trained word embeddings created from *Gensim* and *GloVe* as an input to *Embedding layer* as shown in Fig. 3. We performed our experiment on both embeddings separately. The embedded input was then given as an input to the proposed model. We started our experiment with only one layer of convolution (1-CNN) without any regularization (or Dropout) layer for the review helpfulness prediction task. The MSE for 1-CNN without Dropout with *Gensim* embedding was 1.562 and 2.258 for *Amazon* and *Snapdeal* respectively. Whereas, for *GloVe* embedding the MSE was 0.248 and 0.386 for *Amazon* and *Snapdeal* respectively. The MSE was slightly decreased from 1.562 to 1.527 and from 2.258 to 2.226 for *Amazon* and *Snapdeal* after including Dropout layer with 1-CNN for *Gensim* embeddings. The similar effect was found for *GloVe* also for 1-CNN with Dropout layers for both *Amazon* and *Snapdeal* datasets. We repeated the experiments with two layers of convolution (2-CNN) with or without Dropout layers. The best results we got for *GloVe* embedding 2-CNN with Dropout layer as system MSE was 0.213 and 0.223 for *Amazon* and *Snapdeal* respectively.

We plotted the progress of MSE with respect to the number of epochs for both training and testing cases to capture the behaviour of the system for both 1-CNN and 2-CNN without and with Dropout layers in Fig. 5, 6, 7, 8, 9, 10, 11, and 12 respectively. Although the MSE ranges for all cases were different, we have normalized them in the range of 0 to 1 for comparison purpose. As can be seen from Fig. 5, 6 and 7, 8 for 1-CNN model MSE in training case was almost constant after 100 epochs. But, in testing case, MSE starts increasing as the number of epochs increases. Hence we added another layer of convolution and performed the experiments with the 2-CNN model. The graph we obtained for 2-CNN without and with Dropout layer can be seen in Fig. 9, 10, and 11, 12. We plotted the graph between MSE and number of epochs. Here the blue line is for training dataset and the green line is for the testing dataset. As we observe, training performance is continuously increasing as we increase the number of epochs. But, for testing cases, the performance starts increasing as we increase the

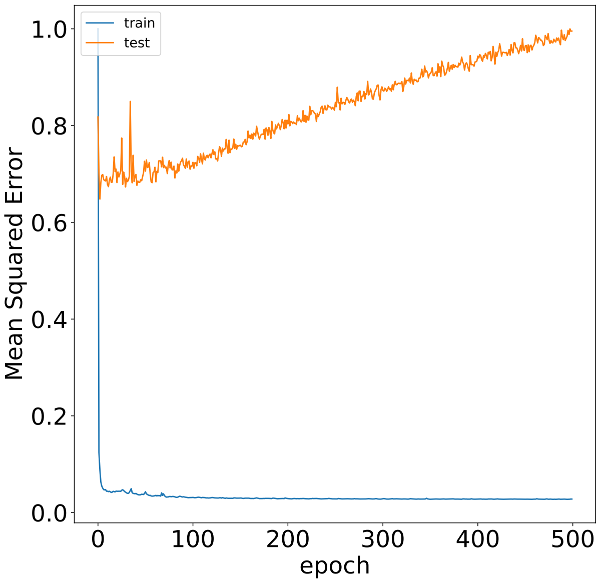


Figure 5: MSE Vs epochs for 1-CNN without Dropout layer using GloVe embedding for Amazon Products

epochs but, after a certain point it starts gradually decreasing. We captured that point when the MSE value was minimum and found that in all cases 100 epochs are enough to validate the proposed model. Later, we performed all the testing for 100 epochs.

In the proposed 2-CNN model, the system learned weights of filters. We used two layers of filters after each convolution. Three filter regions (as discussed in Section 3) of size tri-gram, four-gram and five-gram were used to map feature sets from convolution. We started our experiments with one filter region at a time. First, we used tri-gram as a filter with size 3 × 100 where 3 represents the three rows of tri-gram filters and 100 represents the dimension of filters. The filter then started convolving on 50 × 100 size sentence representation of reviews which yielded in feature sets. Here, 50 is a number of words in a review and 100 is the number of dimensions in which each word is represented. We used 128 such filters of tri-grams. After two layers of convolution and max-pooling, the inputs were then fed to a fully connected layer which produced the results at the output layer. For the predicted score then we calculated loss using Equation 10. The calculated loss were then back-propagated to the system using Equations 11, 12, and 13. We iterated this process 100 times and stored the final score of reviews. Similarly, we performed experiments for four-gram filter regions and five-gram filter regions. Finally, we combined all three filter regions and tested our model on that. The reported MSE loss for five different products of *Amazon* and *Snapdeal* is discussed below. The results were obtained for testing datasets with 100 epochs and 128 batch size.

Table 4 shows the results of *Amazon* datasets. As can be seen from Table 4, five individual experiments were conducted for five different products and finally the reviews of all products of *Amazon* were fed as an input to the system. The results for each category have listed in Table 4 separately. As we observe, in each category of product the system performed best when we convolved with all three filter regions. For example, the MSE for *Baby product* was 0.564 when we performed convolution only with tri-gram

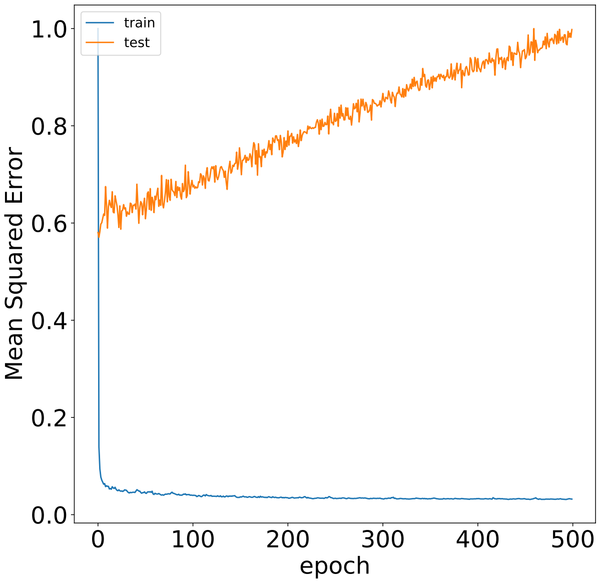


Figure 6: MSE Vs epochs for 1-CNN without Dropout layer using GloVe embedding for snapdeal products

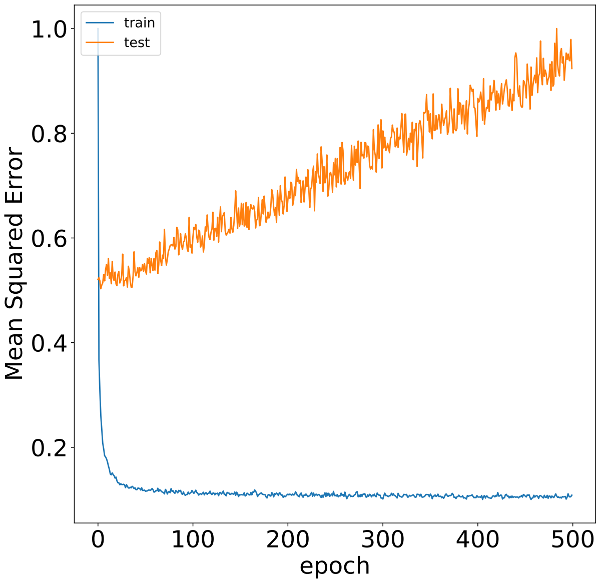


Figure 7: MSE Vs epochs for 1-CNN with Dropout layer using GloVe embedding for Amazon Products

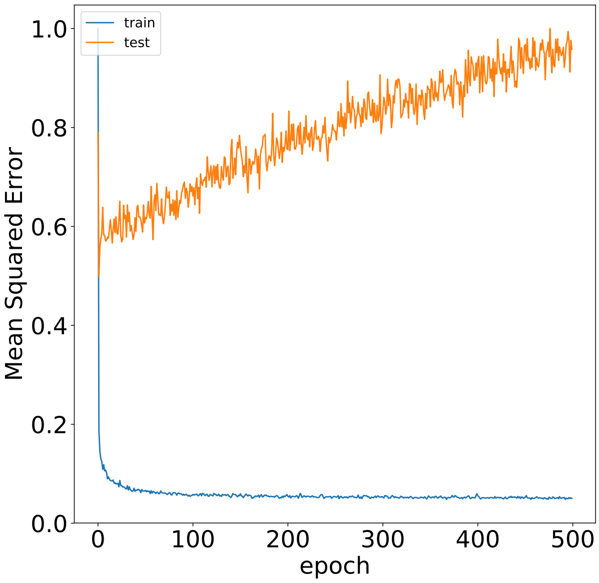


Figure 8: MSE Vs epochs for 1-CNN with Dropout layer using GloVe embedding for snapdeal products

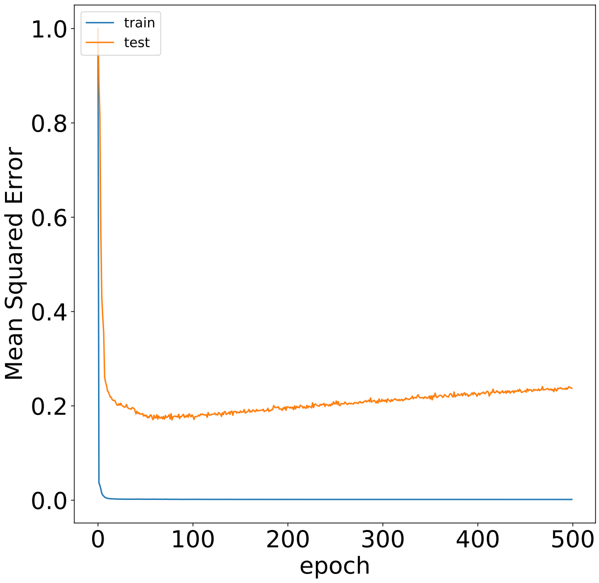


Figure 9: MSE Vs epochs for 2-CNN without Dropout layer using GloVe embedding for Amazon Products

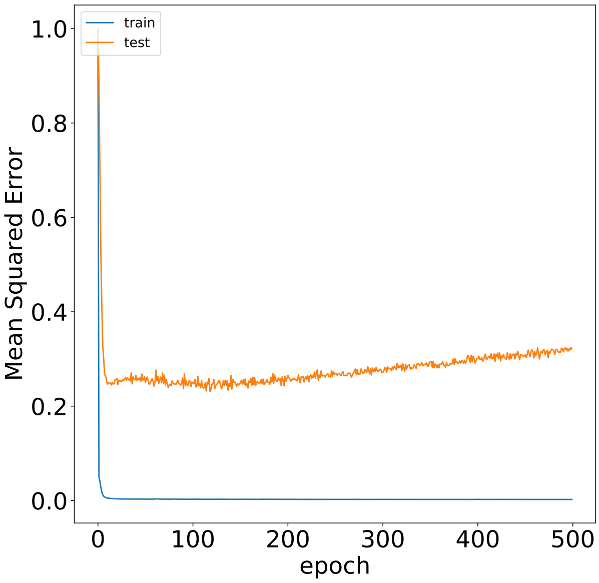


Figure 10: MSE Vs epochs for 2-CNN without Dropout layer using GloVe embedding for snapdeal products

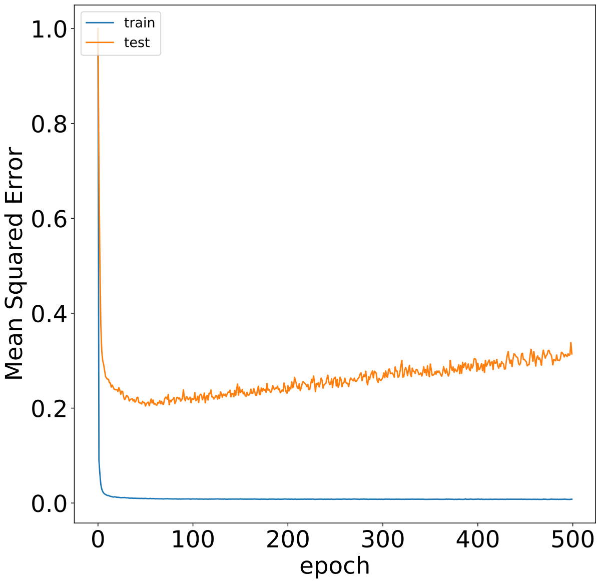


Figure 11: MSE Vs epochs for 2-CNN with Dropout layer using GloVe embedding for Amazon Products

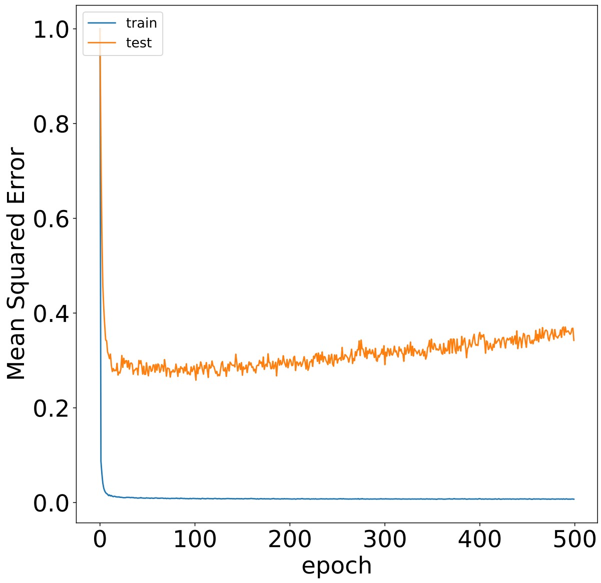


Figure 12: MSE Vs epochs for 2-CNN with Dropout layer using GloVe embedding for snapdeal products

filter regions. The MSE was 0.613 and 0.568 when the convolution was performed on four-gram filter regions and five-gram filter regions respectively. But, the MSE was least as 0.210 when we convolved our input matrix with all three filter regions together. We got similar results for other product categories also. In all cases, the system performed best when we used all three filter regions together. In best cases, for *mobile phones* the MSE was 0.212, for *powerbank* it was 0.199, for *book* 0.289, for *memory card* 0.276 and for all products of *Amazon* it was 0.213.

We performed similar experiments for *Snapdeal* products also. The proposed 2-CNN model was first evaluated with five different products of *Snapdeal* separately and finally, reviews of all five products were fed together to the system. For each category of products, four experiments were conducted. Three individual experiments with three filter regions and one experiment with combined filter regions. As it was the case of *amazon*, for *Snapdeal* products also the system performed best with combined features of tri-gram, four-gram, and five-gram. The MSE for *Baby product* was 0.079, for *mobile phone* it was 0.936, for *powerbank* 0.793, for *book* 0.847, for *memory card* 0.744, and for all products MSE was 0.223.

5 Discussion and Implications

The results of this study showed the effectiveness of incorporating two layers of convolution (2-CNN) in review representation. We also found other models like 1-CNN and 1-CNN with dropout do not perform well compared to 2-CNN with or without dropout. The primary reason was overfitting. 1-CNN with or without dropout got very low MSE on training data but, high MSE on test data. For example, as it is shown in Fig. 5, 6 and 7, 8 the training MSE is less than 0.1 but, the testing MSE starts increasing with

Table 4: 2-CNN results for Amazon products

Approache	**Amazon Product**	Filter Region			**MSE**
		Tri-gram	**Four-gram**	**Five-gram**	
2-CNN	Baby Product	✓	×	×	0.564
		×	✓	×	0.613
		×	×	✓	0.568
		✓	✓	✓	**0.210**
	Mobile Phone	✓	×	×	1.623
		×	✓	×	1.557
		×	×	✓	1.601
		✓	✓	✓	**0.212**
	Power Bank	✓	×	×	2.115
		×	✓	×	2.089
		×	×	✓	2.012
		✓	✓	✓	**0.199**
	Book	✓	×	×	1.362
		×	✓	×	1.174
		×	×	✓	1.161
		✓	✓	✓	**0.289**
	Memory Card	✓	×	×	0.643
		×	✓	×	0.655
		×	×	✓	0.662
		✓	✓	✓	**0.276**
	All products	✓	×	×	0.656
		×	✓	×	1.118
		×	×	✓	1.107
		✓	✓	✓	**0.213**

Table 5: 2-CNN results for Snapdeal products

Approache	Snapdeal Product	Filter Region			MSE
		Tri-gram	**Four-gram**	**Five-gram**	
2-CNN	Baby Product	✓	×	×	0.081
		×	✓	×	0.082
		×	×	✓	0.088
		✓	✓	✓	**0.079**
	Mobile Phone	✓	×	×	2.599
		×	✓	×	2.517
		×	×	✓	2.533
		✓	✓	✓	**0.936**
	Power Bank	✓	×	×	4.013
		×	✓	×	3.991
		×	×	✓	3.874
		✓	✓	✓	**0.793**
	Book	✓	×	×	2.516
		×	✓	×	2.524
		×	×	✓	2.501
		✓	✓	✓	**0.847**
	Memory Card	✓	×	×	4.224
		×	✓	×	4.119
		×	×	✓	4.103
		✓	✓	✓	**0.744**
	All products	✓	×	×	1.316
		×	✓	×	1.310
		×	×	✓	1.287
		✓	✓	✓	**0.223**

epochs. That means, for the proposed review datasets, a neural network-based model with one layer of convolution is not the right choice. The system performed best with 2-CNN plus dropout layer (Fig. 9, 10, 11, and 12).

Meanwhile, we make an analysis about review helpfulness prediction capacity of 2-CNN on each product of *Amazon* and *Snapdeal.* From Table 4, we find the proposed method performed best on *powerbank* dataset with MSE 0.199 and least on *Book* dataset with MSE 0.289. Similarly in the case of *Snapdeal*, from Table 5, we find that the proposed system performed best when we fed reviews of all products together to the system. The MSE was 0.223. The system performance was poor for *mobile phone* dataset.
Although proposed 2-CNN does not require manually engineered features, it has many hyper-parameters to be optimized. The right choice of hyper-parameters can significantly improve the model's performances. In our case, we tuned four hyper-parameters filter regions, dropout rates, epoch size, and batch size. We did the comparison experiments by ten-fold cross-validation. The results are shown in Table 4 and 5 explains that the proposed system performed best when all filter regions that is tri-gram, four-gram, and five-gram were used together. Hence, convolving review matrix together with all three filter regions captured the semantic information of review more accurately. We tried two different dropout rates 0.2 and 0.25 after each Dense layer and found that MSE is least with dropout rate 0.2.
We compared our result with the results proposed by (Saumya et al., 2018) who also worked on the same dataset for their work as we used in our case. Saumya et al. (2018) used gradient boosting regression approach for ranking reviews of e-commerce websites. They used handcrafted features as an input to regressor for predicting best helpful reviews. They calculated the score for each review in two ways: first, they directly fed their extracted features to Gradient Boosting regressor and second, they classified the reviews into low and high quality using Random Forest classifier and then fed only high-quality reviews to the Gradient Boosting regressor. The detailed comparison between our results and the one got by (Saumya et al., 2018) can be seen in Table 6. As it is shown in Table 6, (Saumya et al., 2018) got MSE 2.545 and 0.267 for *Amazon* datasets and MSE 3.434 and 0.623 for *Snapdeal* datasets. They got their best results when they used a hybrid system of classification and regression. In our case we got MSE 0.213 and 0.223 for *Amazon* and *Snapdeal.* Hence, the proposed 2-CNN model outperformed the system proposed by (Saumya et al., 2018). Our system is better than (Saumya et al., 2018) in various ways: first, the system proposed by (Saumya et al., 2018) was very complex as for each review first they performed classification and then regression. In contrast, we directly calculate the score for each review. Second, they used several handcrafted features which is time consuming, costly and biased. The current system feeds continuous vector representation of raw data to the system which eliminates the feature engineering. Third, unlikely Saumya et al. (2018), the current approach used the contextual approach which preserves the semantic information of the review while predicting the best helpful review.

The results of the current study have various implications. First, as per our knowledge, this is the first attempt to predict helpfulness of online reviews using deep convolution representation learning. The convolutional learning removes the need for prior assumptions regarding the functional form or loss distribution as requisite by an earlier

Table 6: Result comparison

Source	Approach	Dataset	MSE
Saumya et al. (2018)	Gradient boosting regression	Amazon	2.545
Saumya et al. (2018)	Random forest classifier + Gradient boosting regression	Amazon	0.267
Saumya et al. (2018)	Gradient boosting regression	Snapdeal	3.434
Saumya et al. (2018)	Random forest classifier + Gradient boosting regression	Snapdeal	0.623
Our case	2-CNN	Amazon	**0.213**
Our case	2-CNN	Snapdeal	**0.223**

approach like linear regression or support vector regression. The presented 2-CNN is capable of preserving complex semantic information of review which was very difficult to obtain from earlier hand-crafted features. Thus, 2-CNN presented in this study minimize the impediment of the existing system by showing the best helpful reviews at front in the review list. This helps the customers to take the full advantage of reviews and make their purchase decision. Moreover, it also removes the obstacle of customers caused by large imbalance immanent in the review quality and sentiment.
Second, using the current research website developers can redesign the listing of reviews. This makes the review ranking system dynamic as even new review get equal chance to appear in the top positions in review list. It will engage more customers and encourage them to write better reviews at any point in the product lifecycle. The current system can be implemented on top of the existing review system of any e-commerce websites. To make easier manipulation of review insertion and rearrangement a separate link list can be created for more helpful reviews. This can be implemented by introducing a new tab say "promising reviews tab" which shows the most useful reviews predicted by the current model to the users as proposed by (Roy et al., 2018) for community question answering domain. The "promising reviews tab" will show only those top k reviews which are predicted by our proposed model. There are some reviews in "promising reviews tab", which are also present in the most helpful reviews tab, however, there are some other new reviews, which are not listed in the most helpful reviews tab. This gives the new high-quality review a fair chance to compete with the old reviews that have obtained votes. The "promising reviews tab" could be integrated with the existing system of online review websites, without any change in internal architecture.

6 Conclusion, Limitation, and Future Direction

We proposed a novel two-layer convolutional neural network model (2-CNN) to learn review representation for predicting best helpful reviews. At first layer, we gave sentence representation of review as an input to convolution. At the second layer, we learned document representation by incorporating sentence weights to the model semantic representation of review. We conducted experiments on our crawled data from *Amazon* and *Snapdeal*. Different variations of proposed model experimented. The results showed that one layer of convolution (1-CNN) was not the right choice for predicting best helpful reviews. In contrast, 2-CNN performed best with dropout layers for both datasets. We also tuned another hyper-parameter, filter-region for tri-gram, four-gram, and five-gram. The best results were obtained when all three regions were used together. The system performance was measured in terms of MSE. The least MSE 0.213 we got for 2-CNN with dropout for *Amazon* dataset.

In the current research dataset choice was restricted to *Amazon* and *Snapdeal*. However, it may possible to use the same methodology for other review websites like Yelp, TripAdvisor. The current work uses only review text as an input to the model and predicts its helpfulness score, which makes it a generalized system. Because it is usual to have the review text field available on the other review websites like Yelp and TripAdvisor. Further, the current model does not use any review metadata information which use to differ on various websites. In the future, other neural network models such as a recurrent neural network can be used. The system performance can be further improved by using some extra features like review metadata features and reviewer features along with the review text.

Acknowledgment

The author would like to acknowledge the Ministry of Electronics and Information Technology (MeitY), Government of India for supporting the financial assistance during research work through “Visvesvaraya PhD Scheme for Electronics and IT”.

Compliance with Ethical Standards

Funding: This study was funded by the Ministry of Electronics and Information Technology (MeitY), Government of India through “Visvesvaraya PhD Scheme for Electronics and IT”.

Conflict of Interest: Author, Sunil Saumya, has received research grants from Ministry of Electronics and Information Technology (MeitY), Government of India through “Visvesvaraya PhD Scheme for Electronics and IT”. Author, Jyoti Prakash Singh, declares that he has no conflict of interest. Author, Yogesh K. Dwivedi, declares that he has no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

Allahbakhsh, M., A. Ignjatovic, H. R. Motahari-Nezhad, and B. Benatallah (2015). Robust evaluation of products and reviewers in social rating systems. *World Wide Web 18*(1), 73–109.

Baek, H., S. Lee, S. Oh, and J. Ahn (2015). Normative social influence and online review helpfulness: Polynomial modeling and response surface analysis. *Journal of Electronic Commerce Research 16*(4), 290.

Bengio, Y., R. Ducharme, P. Vincent, and C. Jauvin (2003). A neural probabilistic language model. *Journal of machine learning research 3*(Feb), 1137–1155.

Bowman, S. R., G. Angeli, C. Potts, and C. D. Manning (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

BrightLocal (2016). Local consumer review survey accssed from www.brightlocal.com/learn/local-consumer-review-survey/ on 22nd december 2016.

Cao, Q., W. Duan, and Q. Gan (2011). Exploring determinants of voting for the helpfulness of online user reviews: A text mining approach. *Decision Support Systems 50*(2), 511–521.

Chen, C., Y. Yang, J. Zhou, X. Li, and F. S. Bao (2018). Cross-domain review helpfulness prediction based on convolutional neural networks with auxiliary domain discriminators. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Volume 2, pp. 602–607.

Chen, H.-N. and C.-Y. Huang (2013). An investigation into online reviewers' behavior. *European Journal of Marketing 47*(10), 1758–1773.

Chua, A. Y. and S. Banerjee (2015). Understanding review helpfulness as a function of reviewer reputation, review rating, and review depth. *Journal of the Association for Information Science and Technology 66*(2), 354–362.

Chua, A. Y. and S. Banerjee (2016). Helpfulness of user-generated reviews as a function of review sentiment, product type and information quality. *Computers in Human Behavior 54*, 547–554.

Chua, A. Y. and S. Banerjee (2017). Analyzing review efficacy on amazon. com: Does the rich grow richer? *Computers in Human Behavior 75*, 501–509.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research 12*(Aug), 2493–2537.

Danescu-Niculescu-Mizil, C., G. Kossinets, J. Kleinberg, and L. Lee (2009). How opinions are received by online communities: a case study on amazon. com helpfulness votes. In *Proceedings of the 18th international conference on World wide web*, pp. 141–150. ACM.

dos Santos, C. and M. Gatti (2014). Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 69–78.

Forman, C., A. Ghose, and B. Wiesenfeld (2008). Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *Information Systems Research 19*(3), 291–313.

Freixas, X., R. Guesnerie, and J. Tirole (1985). Planning under incomplete information and the ratchet effect. *The review of economic studies 52*(2), 173–191.

Gao, B., N. Hu, and I. Bose (2017). Follow the herd or be myself? an analysis of consistency in behavior of reviewers and helpfulness of their reviews. *Decision Support Systems 95*, 1–11.

Ghose, A. and P. G. Ipeirotis (2006). Designing ranking systems for consumer reviews: The impact of review subjectivity on product sales and review quality. In *Proceedings of the 16th annual workshop on information technology and systems*, pp. 303–310.

Ghose, A. and P. G. Ipeirotis (2011). Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering 23*(10), 1498–1512.

Graham, B. (2014). Fractional max-pooling. *arXiv preprint arXiv:1412.6071*.

Guo, B. and S. Zhou (2017). What makes population perception of review helpfulness: an information processing perspective. *Electronic Commerce Research 17*(4), 585–608.

Hu, N., N. S. Koh, and S. K. Reddy (2014). Ratings lead you to the product, reviews help you clinch it? the mediating role of online review sentiments on product sales. *Decision support systems 57*, 42–53.

Huang, A. H., K. Chen, D. C. Yen, and T. P. Tran (2015). A study of factors that contribute to online review helpfulness. *Computers in Human Behavior 48*, 17–27.

Kalchbrenner, N., E. Grefenstette, and P. Blunsom (2014). A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Kaushik, K., R. Mishra, N. P. Rana, and Y. K. Dwivedi (2018). Exploring reviews and review sequences on e-commerce platform: A study of helpful reviews on amazon. in. *Journal of Retailing and Consumer Services 45*, 21–32.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kizgin, H., A. Jamal, B. L. Dey, and N. P. Rana (2018). The impact of social media on consumers' acculturation and purchase intentions. *Information Systems Frontiers 20*(3), 503–514.

Kohavi, R. et al. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, Volume 14, pp. 1137–1145. Montreal, Canada.

Korfiatis, N., E. García-Bariocanal, and S. Sánchez-Alonso (2012). Evaluating content quality and helpfulness of online product reviews: The interplay of review helpfulness vs. review content. *Electronic Commerce Research and Applications 11*(3), 205–217.

Krishnamoorthy, S. (2015). Linguistic features for review helpfulness prediction. *Expert Systems with Applications 42*(7), 3751–3759.

Kumar, A., O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher (2016). Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, pp. 1378–1387.

Kumar, A. and J. P. Singh (2018). Location reference identification from tweets during emergencies: A deep learning approach. *International Journal of Disaster Risk Reduction*.

Lee, E.-J. and S. Y. Shin (2014). When do consumers buy online product reviews? effects of review quality, product type, and reviewer's photo. *Computers in Human Behavior 31*, 356–366.

Lee, S. and J. Y. Choeh (2014). Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications 41*(6), 3041–3046.

Levy, O. and Y. Goldberg (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Volume 2, pp. 302–308.

Li, L., B. Qin, W. Ren, and T. Liu (2017). Document representation and feature combination for deceptive spam review detection. *Neurocomputing*.

Li, M., L. Huang, C.-H. Tan, and K.-K. Wei (2013). Helpfulness of online product reviews as seen by consumers: Source and content features. *International Journal of Electronic Commerce 17*(4), 101–136.

Liu, J., Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou (2007). Low-quality product review detection in opinion summarization. In *EMNLP-CoNLL*, Volume 7, pp. 334–342.

Liu, Y., X. Huang, A. An, and X. Yu (2008). Modeling and predicting the helpfulness of online reviews. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 443–452. IEEE.

Liu, Y., C. Jiang, and H. Zhao (2018). Using contextual features and multi-view ensemble learning in product defect identification from online discussion forums. *Decision Support Systems 105*, 1–12.

Liu, Y., X. Yu, A. An, and X. Huang (2013). Riding the tide of sentiment change: sentiment analysis with evolving online reviews. *World Wide Web 16*(4), 477–496.

McCollum, P. (1997). An introduction to back propagation neural networks. *The Newsletter of the Seattle Robotics Society*.

Merton, R. K. et al. (1968). The matthew effect in science. *Science 159*(3810), 56–63.

Mikolov, T., M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur (2010). Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119.

Mudambi, S. M. and D. Schuff (2010). What makes a helpful review? a study of customer reviews on amazon. com. *MIS quarterly 34*(1), 185–200.

Nair, V. and G. E. Hinton (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Osgood, D. W., L. L. Finken, and B. J. McMorris (2002). Analyzing multiple-item measures of crime and deviance ii: Tobit regression analysis of transformed scores. *Journal of Quantitative Criminology 18*(4), 319–347.

Otterbacher, J. (2009). 'helpfulness' in online communities: a measure of message quality. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 955–964. ACM.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543.

Prieto, A., B. Prieto, E. M. Ortigosa, E. Ros, F. Pelayo, J. Ortega, and I. Rojas (2016). Neural networks: An overview of early research, current frameworks and new challenges. *Neurocomputing 214*, 242–268.

Qazi, A., K. B. S. Syed, R. G. Raj, E. Cambria, M. Tahir, and D. Alghazzawi (2016). A concept-level approach to the analysis of online review helpfulness. *Computers in Human Behavior 58*, 75–81.

Ren, Y. and D. Ji (2017). Neural networks for deceptive opinion spam detection: An empirical study. *Information Sciences 385*, 213–224.

Roy, P. K., Z. Ahmad, J. P. Singh, M. A. A. Alryalat, N. P. Rana, and Y. K. Dwivedi (2018). Finding and ranking high-quality answers in community question answering sites. *Global Journal of Flexible Systems Management 19*(1), 53–68.

Saini, S., S. Saumya, and J. P. Singh (2017). Sequential purchase recommendation system for e-commerce sites. In *IFIP International Conference on Computer Information Systems and Industrial Management*, pp. 366–375. Springer.

Saumya, S., J. Kumar, and J. P. Singh (2018). Genre fraction detection of a movie using text mining. In *Advanced Computing and Systems for Security*, pp. 167–177. Springer.

Saumya, S. and J. P. Singh (2018). Detection of spam reviews: a sentiment analysis approach. *CSI Transactions on ICT*, 1–12.

Saumya, S., J. P. Singh, A. M. Baabdullah, N. P. Rana, and Y. K. Dwivedi (2018). Ranking online consumer reviews. *Electronic Commerce Research and Applications 29*, 78–89.

Saumya, S., J. P. Singh, and P. Kumar (2016). Predicting stock movements using social network. In *Conference on e-Business, e-Services and e-Society*, pp. 567–572. Springer.

Schumaker, R. P., Y. Zhang, C.-N. Huang, and H. Chen (2012). Evaluating sentiment in financial news articles. *Decision Support Systems 53*(3), 458–464.

Shareef, M. A., Y. K. Dwivedi, V. Kumar, G. Davies, N. Rana, and A. Baabdullah (2018). Purchase intention in an electronic commerce environment: A trade-off between controlling measures and operational performance. *Information Technology & People*.

Siering, M. and J. Muntermann (2013). What drives the helpfulness of online product reviews? from stars to facts and emotions. In *Wirtschaftsinformatik*, pp. 7.

Siering, M., J. Muntermann, and B. Rajagopalan (2018). Explaining and predicting online review helpfulness: The role of content and reviewer-related signals. *Decision Support Systems 108*, 1–12.

Singh, J. P., Y. K. Dwivedi, N. P. Rana, A. Kumar, and K. K. Kapoor (2017). Event classification and location prediction from tweets during disasters. *Annals of Operations Research*, 1–21.

Singh, J. P., S. Irani, N. P. Rana, Y. K. Dwivedi, S. Saumya, and P. K. Roy (2017). Predicting the “helpfulness" of online consumer reviews. *Journal of Business Research 70*, 346–355.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

Spool, J. (2009). The magic behind amazon's 2.7 billion dollar question. available online at http://www.uie.com/articles/magicbehindamazon/2009 (accessed on 15th may 2016).

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research 15*(1), 1929–1958.

Tsao, W.-C. (2014). Which type of online review is more persuasive? the influence of consumer reviews and critic ratings on moviegoers. *Electronic Commerce Research 14*(4), 559–583.

Ullah, R., A. Zeb, and W. Kim (2015). The impact of emotions on the helpfulness of movie reviews. *Journal of applied research and technology 13*(3), 359–363.

Wan, Y. (2015). The matthew effect in social commerce. *Electronic Markets 25*(4), 313–324.

Wan, Y., B. Ma, and Y. Pan (2018). Opinion evolution of online consumer reviews in the e-commerce environment. *Electronic Commerce Research 18*(2), 291–311.

Wan, Y. and M. Nakayama (2014). The reliability of online review helpfulness. *Journal of Electronic Commerce Research 15*(3), 179.

Wang, D., S. Zhu, and T. Li (2013). Sumview: A web-based engine for summarizing product reviews and customer opinions. *Expert Systems with Applications 40*(1), 27–33.

Wang, Z. and A. C. Bovik (2009). Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine 26*(1), 98–117.

Weathers, D., S. D. Swain, and V. Grover (2015). Can online product reviews be more helpful? examining characteristics of information content by product type. *Decision Support Systems 79*, 12–23.

Wu, J. (2017). Review popularity and review helpfulness: A model for user review effectiveness. *Decision Support Systems 97*, 92–103.

Wu, J., Y. Wu, J. Sun, and Z. Yang (2013). User reviews and uncertainty assessment: A two stage model of consumers' willingness-to-pay in online markets. *Decision Support Systems 55*(1), 175–185.

Zhang, Y. and Z. Lin (2018). Predicting the helpfulness of online product reviews: A multilingual approach. *Electronic Commerce Research and Applications 27*, 1–10.

Sunil Saumya. National Institute of Technology Patna, India. *E-mail address*: sunils.cse15@nitp.ac.in

Jyoti Prakash Singh. National Institute of Technology Patna, India. *E-mail address*: jps@nitp.ac.in

Yogesh K. Dwivedi. Swansea University Bay Campus, Swansea, UK. *E-mail address*: ykdwivedi@gmail.com