From Helpfulness Prediction to Helpful Review Retrieval for Online Product Reviews

Chau Vo Dung Duong Duy Nguyen Tru Cao

Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology
Vietnam National University – Ho Chi Minh City
Ho Chi Minh City, Vietnam
chauvtn@hcmut.edu.vn, tridungduong16@gmail.com, ntkduy123@gmail.com, tru@hcmut.edu.vn

ABSTRACT

Nowadays, online product reviews belong to a valuable data source for customers in e-commerce. They provide customers with helpful details about a given product before customers make a decision on purchasing that product. Nevertheless, in this regard, if the e-commerce system returns too many reviews to customers and the reviews are not well presented in a relevant manner, the reviews might become cumbersome and time-consuming. In this paper, we define a helpful review retrieval task to support the customers by returning a ranked list of helpful reviews according to their helpfulness about the product of their interest. For an effective solution to the task, we also propose a method with an enhanced list of features for review representation and a multiple linear regression model using the elastic net regularization method. Our method is comprehensive as examining the task in its entirety from review's helpfulness prediction to helpful review retrieval for online product reviews. Evaluated on a real world Amazon dataset of the reviews about electronic devices, our method outperforms the others with the best values: 0.8 for the Normalized Discounted Cumulative Gain measure and 0.83 for the Accuracy measure. Such promising experimental results confirm the effectiveness of our method for the task.

CCS CONCEPTS

• Information systems ~ Information retrieval ~ Retrieval models and ranking ~ Learning to rank • Information systems ~ Information retrieval ~ Evaluation of retrieval results ~ Relevance assessment; Retrieval effectiveness

SoICT 2018, December 6-7, 2018, Danang City, Vietnam
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-6539-0/18/12...$15.00
https://doi.org/10.1145/3287921.3287931

KEYWORDS

Review helpfulness prediction, Helpful review retrieval, Feature extraction, Multiple linear regression model with the elastic net regularization method, Normalized Discounted Cumulative Gain

ACM Reference format:

Chau Vo, Dung Duong, Duy Nguyen and Tru Cao. 2018. From Helpfulness Prediction to Helpful Review Retrieval for Online Product Reviews. *In SoICT '18: Ninth International Symposium on Information and Communication Technology, December 6–7, 2018, Danang City, Viet Nam.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3287921.3287931.

1 INTRODUCTION

In the current digital age, e-commerce has been developed very well all over the world. A large number of customers purchase many various products in the online market. The online market is famous for its non-face-to-face transactions. Therefore, in such a market, information is a valuable source for the customers so that they can make an appropriate decision about the product. In the past, the customers had to gather the information about a product via different channels on their own. Nowadays, information and knowledge sharing is encouraged ubiquitously. After purchasing a product, the customers will be asked about the product and sometimes they themselves would like to give feedbacks about the product and the commercial transaction they have made. Online product reviews become popular in many e-commerce systems.

As a result, those reviews turn into hidden advisors of new potential customers of those products in the future. This is because the new potential customers may find them helpful for supporting their decision making on purchasing a related product. Nevertheless, in this regard, if the e-commerce system returns too many reviews to customers and they are not well presented in a relevant manner, those reviews might become cumbersome and time-consuming. Therefore, we address the problem of helpful review retrieval for online product reviews in e-commerce.

Aware of the value of those online product reviews, their helpfulness has been analyzed for supporting the customers more and more. Many works have been proposed with many various learning approaches for many different objectives. Some of them in [8, 11-14, 17, 19, 20, 22] followed the supervised learning approach with regression models and some of them in [4-6, 15, 18, 19] with classification models, while some of them in [9, 10, 16] used the unsupervised learning approach.

Using regression models, the existing works predicted the helpfulness values of new reviews. For regression model construction, many methods have been used: linear regression [17, 19, 22], support vector regression [8, 22], neural networks [12], and ensembles [11, 13, 14, 19]. Using classification models, the existing works classified a new review into either "helpful" or "non-helpful" class. As for classification, many various models have been built: Naïve Bayes [18, 19], support vector machines [4, 18, 19], random forests [4, 6, 19], decision trees [5], and deep neural networks [15]. For the unsupervised approach, Latent Dirichlet Allocation has been used in [10, 16].

Along with helpfulness prediction models, many various lists of features have been proposed in the existing works. It is realized that the feature list and model are different from work to work with certain effectiveness on many various datasets for different products. Only [18] and ours are based on the same datasets about electronic devices at Amazon.com, while the others used different datasets prepared and crawled by themselves. Such a situation reflects the active research on this problem.

Based on the output from each model, several works [6, 8, 9, 10, 16, 19] further considered review rankings, while some of them [4, 12, 15] analyzed the influence factors on the helpfulness of a review. For review rankings that must be included in helpful review retrieval, only [19] is somewhat similar to our work, while the others did not actually perform *helpful review* ranking. However, the method proposed in [19] is complicated for helpful review retrieval with a classification model for helpful reviews and then a regression model for helpfulness prediction and further for ranking. In addition, only a *top k-review* list was evaluated for true matching and returned. It is believed that a full ranked list of helpful reviews should be returned as the output to customers. Moreover, not only truly matched reviews but also the correct orders of helpful reviews in the list are required in the problem.

From the existing works reviewed above, we figure out that helpful review retrieval has not yet been a solved problem. The problem needs a more comprehensive task definition and corresponding solution. Therefore, in this paper, we first define a helpful review retrieval task to support the customers by returning a ranked list of helpful reviews according to their helpfulness about the product of their interest. For an effective solution to the task, we also propose a method with an enhanced list of features for review representation and a multiple linear regression model using the elastic net regularization. Our method is comprehensive as examining the task in its entirety from review's helpfulness prediction to helpful review retrieval for online product reviews. Evaluated on a real world Amazon dataset of the reviews about electronic devices, our method outperforms the others with the best values: 0.8 for the Normalized Discounted Cumulative Gain measure and 0.83 for the Accuracy measure. Such promising experimental results confirm the effectiveness of our method.

The rest of this paper is structured as follows. In section 2, we define a helpful product review retrieval task. In section 3, we propose a method to support this task and compare our method with the existing ones from the theoretical perspectives. Section 4 is then presented for our empirical evaluation study. In section 5, our work is concluded and its future extensions are stated.

2 TASK DEFINITION

Helpful product review retrieval is significant for e-commerce to return a relevantly ranked list of the most helpful product reviews as per a customer's query about a specific product. There are two reasons for this significance. For the first reason, it is hard for the customer to consume too many returned reviews regardless of their helpfulness. For the latter, a ranked list is obviously better than a non-ranked one when the customer checks the returned reviews one by one in a certain order.

In this paper, we define this helpful product review retrieval task in a comprehensive manner from helpfulness prediction to helpful product review retrieval with helpfulness-based ranking. It is somehow related to helpfulness prediction tasks in [1].

The input of this task is a given list of reviews about a specific product. This review list can be a result of some filtering or searching process in an e-commerce system. Each review is regarded as a document in information retrieval, including: review identifier, product identifier, reviewer's name, the number of helpful votes, the total number of all the votes, overall review text, detailed review text, review summary text, review date and time.

The output of this task is a ranked list of helpful reviews.

With the input and output defined above, in practice, an e-commerce system can return a ranked helpful review list to a customer once this customer views a detailed page of a particular product. Therefore, the process of this task includes two key phases: determine which reviews are helpful and rank the helpful reviews according to their helpfulness. The first phase is in fact the helpfulness prediction subtask while the second one is the ranking subtask with respect to a specific criterion, which is helpfulness. The helpfulness prediction subtask can be resolved by machine learning approaches. In our work, the supervised learning approach is used. The second one is trivial with any existing sorting algorithm as soon as helpfulness of each review is available. For ranking, helpfulness values must be ordinal.

For more clarity, we include some definitions as follows.

In [8], helpfulness of a voted review about a product is a proportion of helpful votes to all the votes given to a review. A higher helpfulness value is preferable to customers in supporting them to make a better decision about purchasing that product.

Based on such helpfulness values, we further define a helpful review as a review with a helpfulness value greater than or equal to a given threshold. Like other existing works [4, 15, 18], in our work, a predefined value 0.6 is used for this threshold.

A ranked list of helpful product reviews is a list where the more helpful reviews are placed before the less ones.

3 THE PROPOSED METHOD

This section is dedicated to our method, which is proposed as a solution to the aforementioned task. Its details are described in subsections. A brief comparison with the existing methods is also included to highlight the contributions of the proposed method.

3.1 An Overview

Figure 1 is sketched to give an overview on the proposed method. Shown in this figure, our method includes two main steps: (1). Model Construction and (2). Review Retrieval.

In the first step, using the available reviews along with their helpfulness values, a helpfulness prediction model is constructed in the supervised learning manner. These reviews are preprocessed and their feature values are extracted by the Feature Extraction module. Their corresponding vectors in the computational form are achieved and labeled with the helpfulness values. All of them are input to a learning algorithm to learn a prediction model. This resulting model is then validated until its effectiveness is high enough for use in the next step.

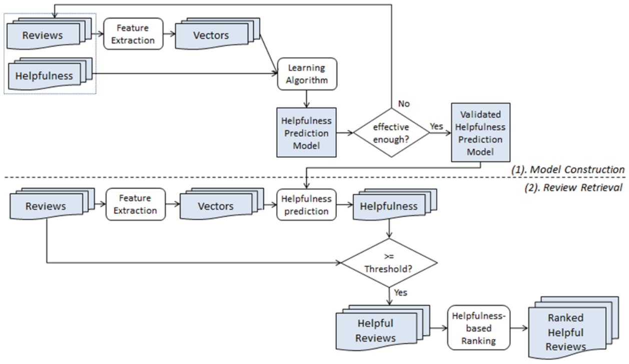

Figure 1: An Overview of the Proposed Method.

In the second step, each new review is also preprocessed and then its feature values are extracted to generate a corresponding vector. This vector is input to the validated helpfulness prediction model to predict a helpfulness value of the input review. The resulting helpfulness value is checked with the threshold. If it is greater than or equal to the threshold, the input review is considered "helpful" and included into the ranked list and returned to customers.

In this overview, we excluded the preprocessing subtasks related to the natural language processing techniques for simplicity. They are: stemming for converting words to their roots; removing punctuations such as comma, semicolon, and period; and removing stopwords. These preprocessing subtasks are done before the Feature Extraction module is performed.

3.2 Feature Extraction

In the proposed method, the Feature Extraction module is an important component. It helps generating a vector in the computational form for each review so that the remaining process can be conducted on those vectors instead of the texts in the reviews. The effectiveness of the remaining process is also influenced by this module if the details of each review can not be captured well enough from many various aspects.

Due to its importance, we first design an initial feature list. After that, we added two features more as a contribution of our method to the task. The initial feature list is based on that in [18] while the added features are from inherent differences between helpful and non-helpful reviews used in practice. All the features in our method are described as follows:

3.2.1 Anatomical Feature Group. We call this group AG. In this group, the features are related to the structure of each review text. They are listed below:

- *Average sentence length*: this feature gives us an average sentence length in a review. A review with too short or long sentences might not be a helpful review. Therefore, this feature helps excluding non-helpful reviews with extreme sentence lengths.
- *The number of sentences*: this feature gives us the length of a review. If written with so few sentences, a review might not contain enough details about a product. Therefore, it helps identifying helpful reviews.
- *The number of characters*: this feature is similar to the previous one but computed with detailed information at the character level.
- *The number of exclamation and question marks*: due to the nature of a review, exclamation and question marks are rarely used for comments about a product. Therefore, a review with many exclamation and question marks might be a non-helpful review.
- *The number of capitalized words*: similar to the previous feature, if containing too many capitalized words, a review tends to be a non-helpful review. This is because capitalized words have a special use context while review text is used for descriptions and assessments about a product.

3.2.2 Meta-data Feature Group. We call this group MG. Different from AG, MG includes the features not related to review text. Instead, they are related to extra information about a review. There are two MG features.

- *The number of stars*: if a review can receive more stars (i.e., higher rating scores), its helpfulness is recognized more. Therefore, this feature can reflect the helpfulness of each review. Its value stems from the details of each review straightforwardly.
- *Deviation from popular opinion*: this feature is derived from the existing ratings that all the customers gave to a review. It is calculated as the difference between the value of the previous feature and its averaged value for each review. Although the previous feature, "*the number of stars*", is useful for recognizing a helpful review, extreme values of the previous feature might lead to misrecognize a non-helpful review. Therefore, this feature is added to distinguish non-helpful reviews from helpful ones.

3.2.3 Lexical Feature Group. We call this group LG. The LG features are related to the lexical aspect of review text.

- *tf-idf for unigrams*: the importance of each unigram in a review is captured to represent the review. If a review contains more unigrams with less importance, it appears to be non-helpful. The *tf-idf* formula can be found in [18].
- *Readability*: considered in the task's context, a review is more helpful if its readers can consume it more conveniently. Therefore, its readability must be high so that its content can be processed well, leading to its helpfulness increase. Similarly, its formulas can be found in [18].

3.2.4 Added Feature Group. In this group, two new features are added below:

- *The number of helpfulness votes*, called HV#: the value of this feature can let us know if the review is new or not, if the review has been evaluated by any customers, and if the review has been considered "helpful" by any customers. It belongs to the MG group. The higher value of this feature implies the more helpful review. Therefore, intuitively this feature can help discriminating between reviews according to their helpfulness.
- *The number of positive/negative words*, called PN#: according to the task's context, a helpful review normally contains comments about a given product. More detailed comments need more positive/negative words to express a reviewer's opinions. Therefore, this feature can reflect the differences between helpful and non-helpful reviews. It examines the semantics of a review. The larger value of this feature implies the more helpful review.

In summary, the initial feature list includes 3 feature groups (AG, MG, LG). The final enhanced list consists of 3 feature groups (AG, MG, LG) and two new features (HV#, PN#). With these two additional features, we expect to improve the effectiveness of the proposed method for helpfulness prediction.

3.3 Helpfulness Prediction with a Regression Model

In this subsection, we introduce a regression model built in the supervised learning manner for helpfulness prediction. The reason for a regression model instead of a classification model is that we would like to estimate the helpfulness of each review as a numeric value in the finest range [0, 1]. That range enables us to do ranking in more detail as compared to a set of nominal values {"Helpful", "Non-helpful"} predicted by a classification model.

As our work in its infancy, a multiple linear regression model is built in this step. We chose this regression model for its simplicity and efficiency. Furthermore, its more effective regularized version using the elastic net method proposed in [23] is considered instead of the original simple one.

Let R be a list of reviews about a given product, H be their helpfulness values, X be feature vectors generated from R after the feature extraction substep, and *p* be the number of features that represents each review in the vector space model. The relationship between the reviews and their helpfulness values is expressed as a function *f* such that: H = *f*(X). This function *f* can be learnt by many various supervised learning algorithms for a regression model. Once *f* is learnt, we obtain a helpfulness prediction model: *Helpfulness_Prediction_Model* = *f*.

Previously introduced, *f* is a multiple linear regression model with *p* predictors corresponding to *p* features and one response corresponding to helpfulness. The relationship is re-written below:

$$H = \beta_0 + X_1\beta_1 + \ldots + X_p\beta_p + \varepsilon \qquad (1)$$

where $\beta=(\beta_0, \beta_1, \ldots, \beta_p)$ is the regression coefficient vector of the model and ε is the residual vector.

The traditional learning process of this model aims at estimating the values of the coefficients β by minimizing the residual sum of squares:

$$\hat{\beta} = \operatorname{argmin}_{\beta} \left\| H - \beta_0 - X_1\beta_1 - \ldots - X_p\beta_p \right\|^2 \qquad (2)$$

For a more effective model, the elastic net regularization method in [23] was introduced by combining the L_1 and L_2 penalizations of the lasso and ridge methods, respectively. With L_1, it can enable a sparse model in a high-dimensional space, while with L_2, it helps removing the limitation on the number of selected predictors, encouraging grouping effect, and stabilizing the L_1 regularization path. These characteristics are necessary for a review's helpfulness prediction model in our task where data regression is made with many features in a high-dimensional space. Using the elastic net method, the estimates are now redefined as follows:

$$\hat{\beta} = \operatorname{argmin}_{\beta} \left(\left\| H - \beta_0 - X_1\beta_1 - \ldots - X_p\beta_p \right\|^2 + \lambda_2 \left\| \beta \right\|^2 + \lambda_1 \left\| \beta \right\|_1 \right) \qquad (3)$$

Where λ_1 and λ_2 are two non-negative parameters for L_1 and L_2, respectively.

Although a linear regression model, our regression model with the elastic net regularization is expected to be better than a traditional linear regression model and the one with the lasso regularization when applied to the helpfulness prediction step. Our work is also the first one exploiting this regression model using the elastic net method for the helpful review retrieval task.

```
Algorithm: Helpfulness Prediction Model Building
Input:
- R: a list of reviews
- H: Helpfulness values of the reviews in R
Output:
- Helpfulness_Prediction_Model: a helpfulness prediction
  model
Process:
Error = 0.5
Repeat
   X = Feature_Extraction (R)
   Helpfulness_Prediction_Model = ElasticNet (X, H)
   Current_Error = Validate (Helpfulness_Prediction_Model)
   if (Current_Error<Error) Error = Current_Error
until Error is acceptable
return Helpfulness_Prediction_Model
```

Figure 2: Pseudo Code of the Helpfulness Prediction Model Building Step.

Using the elastic net method, this step is conducted for a more effective helpfulness prediction model with its pseudo code in Figure 2. In this algorithm, *Feature_Extraction*() is a function that generates feature vectors for the reviews input in R; *ElasticNet*() is a regression function that builds an Elastic Net model, named *Helpfulness_Prediction_Model*, on the feature vectors X and their corresponding helpfulness values H; and *Validate*() is a function that helps evaluating the resulting *Helpfulness_Prediction_Model* model before used in the next step.

3.4 Helpful Review Retrieval with Helpfulness-based Ranking

In this step, we predict helpfulness of each review and filter helpful reviews using the threshold *T*. Helpful reviews are determined according to their helpfulness values, which must be higher than or equal to a given threshold *T*. They are then sorted in descending order of helpfulness and returned in a ranked list. In our work and the existing ones in [4, 15, 18], this threshold is set to 0.6. Differently, [19] used the averaged helpfulness value as a threshold.

Although simple, this step is included to complete our solution to the helpful review retrieval task. In addition, this step shows that our work actually performs helpfulness-based helpful review ranking.

The pseudo code of this step is summarized in Figure 3. In this algorithm, *Helpfulness_Rank*() is a sorting function that sorts our resulting helpful reviews with respect to their helpfulness.

```
Algorithm: Helpful Review Retrieval
Input:
- R: a list of reviews
- Helpfulness_Prediction_Model: a helpfulness prediction
  model
- T: a threshold
Output:
- Ranked_List: a ranked list of helpful reviews
Process:
X = Feature_Extraction (R)
H = Helpfulness_Prediction_Model (X)
Helpful_R = Ø
for (each review $R_i$ in R)
    if ($H_i$>=T) Helpful_R = Helpful_R U {$R_i$}
end for
Ranked_List = Helpfulness_Rank (Helpful_R)
return Ranked_List
```

Figure 3: Pseudo Code of the Helpful Review Retrieval Step.

3.5 Comparing the Proposed Method with the Existing Ones

From the theoretical perspectives, as a more comprehensive solution to the helpful review retrieval task, our method is different from the existing ones in the following points.

First, the helpful review retrieval task defined in our work is more practical with respect to the customers who require helpful reviews about a given product. In some existing works like [4, 12, 14, 15, 18, 20, 22], the task was considered partially as no helpful review ranking was given. Nevertheless, review ranking has been supported in many existing works. Some of them are [2, 3, 6, 16]. Among them, [2, 3, 6] did not take into account the ranking of helpful reviews according to their helpfulness, while [16] provided helpfulness-based ranking and unfortunately, did not distinguish helpful reviews from non-helpful reviews. Different from these works, for the expected output of the previously defined task, our work returns a ranked list of all the only helpful reviews well presented in descending order of their helpfulness.

Second, our method proposed to use a shorter feature list as compared to a list of observed features on helpfulness summarized in [1]. Based on an initial list in [18], we further examined this list with some other features and found two new features that could improve the performance of the task. Their inherent characteristics show their influence on distinguishing helpful reviews from the others. However, reported in [1], their influence was not clearly recognized for helpfulness prediction. This is understandable because the helpfulness of a review is contextual as investigated in [21]. Therefore, as providing a solution to this task, each existing related work proposed a different feature list and that feature list is different from work to work. In our work, the final enhanced feature list in our method is kept to be simple and effective enough for the task. Our empirical evaluation study will further take it into consideration.

Third, there are many various regression models in the existing works for helpfulness prediction. In [8], support vector regression models were considered for automatically assessing review helpfulness. In [11], an ensemble of the ensembles was built using stochastic gradient boosting and randomized trees. Usefulness of each online review was then predicted by means of an average of the predicted values of the base regression models in the ensemble. In [12], neural networks with the backpropagation learning algorithm were used for helpfulness prediction. In [13], a gradient boosted decision tree model was built in their regression analysis. In [14], a non-linear regression model was built by combining a RBF (radial basis function)-based regression model for the relationship between each individual factor and the helpfulness score. A final helpfulness score is estimated as a weighted sum of its components. In [17], four existing algorithms have been used for solving their regression problem. They are: linear regression, support vector regression, random forest, and M5P. In [19], linear regression and gradient boosting regression models were used to support rank predictions. In [20], a gradient boosting regression model was also used. In [22], linear regression and support vector regression were used for utility scoring, which is somewhat similar to helpfulness prediction.

Except for [8, 19], all the previously mentioned works which proposed regression models for helpfulness prediction did not consider helpful review ranking. Nevertheless, review ranking in [8, 19] is not well fit in our task. In [8], review ranking was generally prepared after helpfulness was determined for each review by regression models. Different from [8], in our work, helpful reviews have been extracted and then ranked to return a ranked *helpful review* list rather than a ranked *review* list. In [19], the authors conducted two rank prediction mechanisms: the first one with classification for helpful review ranking and the latter without classification. Their first one is quite similar to ours. For more details, we found that [19] provided only the top *k*-review list while our method provided a full ranked list of helpful reviews and let the customers decide how many helpful reviews they could examine for the product. Moreover, [19] needed two models: one classification model and one regression model while our method used one regression model with less complexity for the task. In addition, our rankings are evaluated with the Normalized Discounted Cumulative Gain measure for their quality as specified in [1] while [19] used only the matching between the ranked list and the actual one.

Besides the regression models, many different classification models have been examined in the existing works. In [4], support vector machines and random forests were used. In [5], standard decision trees were built for classifying a review into either helpful or non-helpful class. In [6], Random Forests were selected in the experiments. In [15], deep neural networks with 5 layers were built for a binary classification model. In [18], Naïve Bayes and support vector machines with different kernel functions were utilized. In [19], Naïve Bayes, support vector machines, and random forests were used to classify a review as “high-quality review” or “low-quality review” which can be regarded as “helpful review” or “non-helpful review”, respectively.

Except for [6, 19], none of the previous works provided review ranking. In comparison with ours, [6] did not provide helpfulness-based ranking on helpful reviews. Instead, [6] ranked the reviews according to their importance, which is a probability output by a classifier. It can be seen that helpfulness is inherent from the meaning of each review with respect to its users, while such an importance degree from the evaluation of a classifier with respect to the learning process. Therefore, this method is not a suitable solution to our task. For [19], their classification model was used to support helpful review ranking as discussed earlier.

Following the unsupervised learning approach instead of the supervised learning one, the proposed method in [9, 10] was based on Latent Dirichlet Allocation (LDA) and [16] also used LDA and Hidden Markov Model for review rankings. Compared to ours, [9, 10, 16] aimed at a ranked list of reviews from different aspects rather than a ranked list of helpful reviews in our expected output.

Different from the existing works, our work is the first one proposing a multiple linear regression model with the elastic net method for this task. As compared to a traditional multiple linear regression model, our model can provide an elegant, stable, and effective solution to helpfulness prediction. As compared to the proposed models in the aforementioned existing works, our model is less complicated but effective enough for the task. Our work is also extendable as more advanced regression models can be investigated and equipped easily with our proposed method for more effectiveness in practice. In our empirical evaluation, the traditional simple multiple linear regression model is more efficient and becomes a baseline model for comparison. In recent years, it has been compared with the other models in some existing works [12, 14, 17, 19, 22] for helpfulness prediction.

Last but not least, the helpful review retrieval task has been solved for a wide diversity of products. Each product type has many different aspects and properties for which the reviews can be written. Therefore, the datasets used in the proposed works are also different and have their own characteristics, which might lead to different task performances for the proposed methods. In our work, we consider electronic devices in e-commerce of Amazon. Supporting the same product type, [12, 18] had a different focus on helpfulness prediction, while ours on helpful review retrieval.

In short, our work has laid the foundations for the helpful review retrieval task on a given product. More product types can be supported in our practical context for new potential customers.

4 EVALUATION

4.1 Data Descriptions

In this empirical evaluation, a well-known real world Amazon dataset at https://snap.stanford.edu/data/web-Amazon.html was used. We checked 1,241,778 product reviews on electronic devices at Amazon.com. Each review was labeled either “unhelpful” or “helpful”. From the collected large dataset, we randomly extracted both unhelpful and helpful reviews with a balanced scheme. They were then used to prepare the training datasets with different sizes: 500 reviews; 1,000 reviews; 2,000 reviews; 3,000 reviews; and 4,000 reviews. Corresponding to each training dataset, a test dataset was prepared. Each test dataset is ¼ equal to its training dataset. In particular, the test datasets used with different sizes include: 125 reviews; 250 reviews; 500 reviews; 750 reviews; and 1,000 reviews, respectively.

4.2 Experiment Settings

In our evaluation, we examined three following questions:
Question 1: Are the added features effective for the task?
Question 2: Is our method effective for helpfulness prediction of product reviews?
Question 3: Is our method effective for helpful review retrieval with helpfulness-based ranking?

In order to answer these questions, three corresponding experiment groups are prepared as follows. The first one was made to evaluate the addition of the new features. Its experimental results are presented in Table 1. The second one was performed to examine the method for helpfulness prediction with the experimental results shown in Table 2. The last one was conducted for helpful review retrieval with helpfulness-based ranking. Its experimental results are displayed in Table 3. The best results in these tables are recorded in bold.

In the first two experiment groups, Accuracy is used to show how well the method can recognize helpful reviews from a given set of online reviews. In the third one, Normalized Discounted Cumulative Gain (NDCG) is calculated to reflect how well the resulting helpful reviews are ranked in the resulting list for helpful review retrieval as compared to the gold list. Its formula can be seen in [7]. The higher values of the two measures imply the better methods. The two measures, Accuracy and NDCG, were chosen in our evaluation study due to their popularity for classification and ranking, respectively. Especially NDCG is a typical measure for ranking as mentioned in [1].

For comparison in helpfulness prediction with binary classification, we used Naïve Bayes and Support Vector Machines (SVM) with different kernel functions such as linear function (Linear), sigmoid function (Sigmoid), radial basis function (RBF), and polynomial function (Poly). They were chosen according to the evaluation on the same data set as studied in [18]. For comparison in helpfulness prediction with regression, we used linear regression, linear regression with the lasso method, and linear regression with the elastic net method. An estimated numeric value for the helpfulness of each review is compared with the threshold. If it is greater than or equal to the threshold, the review is predicted "helpful"; and otherwise, "unhelpful". A typical value 0.6 is set to the threshold as earlier introduced. For comparison in helpful review retrieval, we also used the previously mentioned regression models. All the predicted "helpful" reviews are ranked according to their estimated numeric helpfulness values. The ranked list is returned to be compared with the real ranked list in the computation of the NDCG measure. All the aforesaid algorithms are from the *sklearn* library with default parameter settings. It is obvious that more advanced algorithms with parameter tuning will provide us with better results than those reported in the next subsection.

4.3 Experimental Results and Discussions

For an answer to the first question, it is realized that in Table 1, most of the cases reported an improvement in accuracy with the added features: "*the number of helpful votes*" (HV#) and "*the number of positive/negative words*" (PN#). Summarized in [1], HV# seems to have no impact on determining the helpfulness of a review. This comment is not always true, depending on training dataset sizes and the classifiers we used. The results with PN# are somehow similar to those with HV#.

However, when we combine them together, the results get more stable. Among 25 cases (5 sizes x 5 models), the final results with 3 feature groups and 2 new features are better than the initial one with 3 feature groups in 20 cases and comparable in 3 cases. Based on the meaningfulness of the added features and the accurate results in about 92% all the cases (23 cases over 25 cases), the added features have made the task more effective.

With the current achievements, we can regard SVM (Linear) and SVM (RBF) trained on a data set of 4,000 reviews as the best models for higher Accuracy (> 80%) in helpfulness prediction. It is noted that the classifiers used in our experiments are typical. More improvement can be reached with more complex classifiers.

For the second question, our observation from the experimental results in Table 2 is the best Accuracy of 83% with the linear regression model (Elastic Net) trained on a dataset of 4,000 reviews. It is also shown the stability of this model when it achieved about 75% or higher than that for correct prediction of the helpfulness of each review. In addition, the comparable capability of a classifier and a regression model in helpfulness prediction on product reviews is remarkable for various sizes of the training datasets. This is understandable because many different approaches have been proposed with either binary classification or regression for this task. As compared previously, many different models have been used and no outstanding prediction model has been confirmed in the existing works.

Table 1: Accuracy Results for Helpfulness Prediction with Different Feature Groups

Size	Feature Group	Naive Bayes	SVM (Linear)	SVM (Sigmoid)	SVM (RBF)	SVM (Poly)
500	3 Initial Feature Groups: AG+MG+LG	0.58	0.53	**0.50**	0.66	**0.66**
	3 Feature Groups + HV#: AG+MG+LG+HV#	0.57	0.55	**0.50**	0.64	**0.66**
	3 Feature Groups + PN#: AG+MG+LG+PN#	0.57	0.55	**0.50**	0.63	0.64
	Final Feature List: AG+MG+LG+HV#+PN#	**0.61**	**0.74**	**0.50**	**0.73**	**0.66**
1,000	3 Initial Feature Groups: AG+MG+LG	0.61	0.50	0.50	0.66	0.66
	3 Feature Groups + HV#: AG+MG+LG+HV#	**0.63**	0.52	0.50	0.66	**0.67**
	3 Feature Groups + PN#: AG+MG+LG+PN#	0.62	0.53	0.50	0.64	0.66
	Final Feature List: AG+MG+LG+HV#+PN#	**0.63**	**0.77**	**0.55**	**0.74**	0.66
2,000	3 Initial Feature Groups: AG+MG+LG	0.60	0.75	0.50	**0.68**	**0.73**
	3 Feature Groups + HV#: AG+MG+LG+HV#	0.63	0.75	0.50	0.67	0.69
	3 Feature Groups + PN#: AG+MG+LG+PN#	0.62	0.75	0.50	0.65	0.67
	Final Feature List: AG+MG+LG+HV#+PN#	**0.68**	**0.77**	**0.55**	0.67	0.68
3,000	3 Initial Feature Groups: AG+MG+LG	0.61	0.50	0.50	0.67	0.68
	3 Feature Groups + HV#: AG+MG+LG+HV#	0.69	0.76	0.50	**0.68**	**0.73**
	3 Feature Groups + PN#: AG+MG+LG+PN#	0.63	0.76	0.50	0.67	**0.73**
	Final Feature List: AG+MG+LG+HV#+PN#	**0.71**	**0.79**	**0.55**	**0.68**	0.72
4,000	3 Initial Feature Groups: AG+MG+LG	0.61	0.77	0.50	0.68	0.69
	3 Feature Groups + HV#: AG+MG+LG+HV#	**0.75**	**0.82**	0.50	0.80	0.76
	3 Feature Groups + PN#: AG+MG+LG+PN#	0.70	0.77	0.51	0.67	**0.78**
	Final Feature List: AG+MG+LG+HV#+PN#	0.74	**0.82**	**0.56**	**0.81**	0.77

As earlier mentioned, the ultimate goal of this task is a ranked list of the reviews that are considered "helpful" to be returned to customers. Therefore, a regression model can give us more information for this goal. As of this moment, a linear regression model can approximate helpfulness values of the helpful reviews similarly predicted by a classifier. This provides us with a "yes" answer to the second question so that our method can be used in the next step for helpful review retrieval.

For the third question, in Table 3, we recorded the NDCG value for each regression model in the case of the training dataset including 4,000 reviews. This is because the most accurate results have been reported for the case of 4,000 reviews in Tables 1 and 2. In this case, the ranked helpful review list created with the linear regression model using the Elastic Net method is the closest to the real helpful review list with the best NDCG value, 0.8, although comparable to the others. The corresponding helpfulness values from the linear regression model with the Elastic Net method can be used to rank the helpful reviews in a more relevant way. Therefore, our method can be an initial effective solution to the helpful review retrieval task.

Table 2: Accuracy Results for Helpfulness Prediction with Different Models

Size	SVM (Linear)	Linear Regression	Linear Regression (Lasso)	Linear Regression (Elastic Net)
500	**0.74**	0.71	0.68	**0.74**
1,000	**0.77**	0.73	0.71	0.76
2,000	**0.77**	0.75	0.74	**0.77**
3,000	**0.79**	0.77	0.75	0.78
4,000	0.82	0.81	0.76	**0.83**

Table 3: NDCG Results for Helpful Review Retrieval on Online Reviews with the Training Data Set of 4,000 Reviews

Model	NDCG
Linear Regression	0.79
Linear Regression (Lasso)	0.77
Linear Regression (Elastic Net)	**0.80**

5 CONCLUSIONS

In this paper, we define a helpful review retrieval task in a more comprehensive manner. After that, we propose a method as a solution to this task. Our method has two main contributions: an enhanced feature list with two new features to represent each review in a computational form and the Elastic Net regression model to predict the helpfulness value of each review and further, to rank the resulting helpful reviews according to their helpfulness. The results on Amazon product reviews were obtained with Accuracy of 83% for helpfulness prediction and with the Normalized Discounted Cumulative Gain measure of 0.8 for helpfulness-based review ranking. Such achievements show that our method is promising for the helpful review retrieval task.

In the future, we want to extend this work in several following perspectives. First, we will reconsider its regression model with other advanced supervised learning algorithms such as deep learning algorithms. Second, representation learning for this task is also interesting and important to improve the effectiveness of a prediction model. Third, more inherent characteristics of reviews and their ranked list will be considered. They are temporal and spatial information associated with each review. Emphasizing their importance, temporal characteristics of reviews were also mentioned in [1]. Finally, we will make more evaluations for many various product categories to bring this work in practice.

REFERENCES

[1] G.O. Diaz and V. Ng, 2018. Modeling and Prediction of Online Product Review Helpfulness: a Survey. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 698-708.

[2] A. Ghose and P.G. Ipeirotis, 2006. Design 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, 1-6.

[3] A. Ghose and P.G. Ipeirotis, 2007. Designing Novel Review Ranking Systems: Predicting Usefulness and Impact of Reviews. In Proceedings of ICEC'07, 1-7.

[4] A. Ghose 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.

[5] M.E. Haque, M.E. Tozal, and M. Islam, 2018. Helpfulness Prediction of Online Product Reviews. In Proceedings of DocEng, 1-4.

[6] J. He, K. Niu, Z. He, S. Wang, and Z. Bie, 2016. A Supervised Method for Ranking Reviews Based on Latent Structure Features. In Proceedings of the 2016 IEEE International Conference on Knowledge Engineering and Applications, 88-92.

[7] K. Järvelin and J. Kekäläinen, 2000. IR Evaluation Methods for Retrieving Highly Relevant Documents. In Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 41-48.

[8] S-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti, 2006. Automatically Assessing Review Helpfulness. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 423–430.

[9] R. Krestel and N. Dokoohaki (June 2015). Diversifying Customer Review Rankings. Neural Networks, 66, 36-45.

[10] R. Krestel and N. Dokoohaki, 2011. Diversifying Product Review Rankings: Getting the Full Picture. In Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, 138-145.

[11] M. Kumar and S. Upadhyay, 2013. Predicting Usefulness of Online Reviews Using Stochastic Gradient Boosting and Randomized Trees. In Proceedings of the 11th Australian Data Mining Conference, 65-72.

[12] S. Lee and J.Y. Choeh (2014). Predicting the Helpfulness of Online Reviews Using Multilayer Perceptron Neural Networks. Expert Systems with Applications, 41, 3041-3046.

[13] B. Li, F. Hou, Z. Guan, A.Y-L. Chong, and X. Pu, 2017. Evaluating Online Review Helpfulness Based on Elaboration Likelihood Model: the Moderating Role of Readability. In Proceedings of the 21st Pacific Aisa Conference on Information Systems, 1-12.

[14] Y. Liu, X. Huang, A. An, and X. Yu, 2008. Modeling and Predicting the Helpfulness of Online Reviews. In Proceedings of the 8th International Conference on Data Mining, 443-452.

[15] M.S.I. Malik and A. Hussain (2017). Helpfulness of Product Reviews as a Function of Discrete Positive and Negative Emotions. Computers in Human Behavior, 1-12. DOI: http://dx.doi.org/10.1016/j.chb.2017.03.053.

[16] S. Mukherjee, K. Popat, and G. Weikum, 2017. Exploring Latent Semantic Factors to Find Useful Product Reviews. In Proceedings of the 2017 SIAM International Conference on Data Mining, 480–488.

[17] Y-J. Part (2018). Predicting the Helpfulness of Online Customer Reviews across Different Product Types. Sustainability, 10(1735), 1-20.

[18] J. Rodak, M. Xiao, and L. Longoria, 2012. Predicting Helpfulness Ratings of Amazon Product Reviews. Technical Project Report, Stanford University, 1-4.

[19] S. Saumya, J.P. Singh, A.M. Baabdullah, N.P. Rana, and Y.K. Dwivedi (2018). Ranking Online Consumer Reviews. Electronic Commerce Research and Applications, 1-25. DOI: https://doi.org/10.1016/j.elerap.2018.03.008.

[20] J.P. Singh, 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.

[21] R. Sipos, A. Ghosh, and T. Joachims, 2014. Was This Review Helpful to You? It Depends! Context and Voting Patterns in Online Content. In Proceedings of WWW'14, 1-11.

[22] Z. Zhang and B. Varadarajan, 2006. Utility Scoring of Product Reviews. In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, 51–57.

[23] H. Zou and T. Hastie (2005). Regularization and Variable Selection via the Elastic Net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2), 301-320.