JOIN (Jurnal Online Informatika)
Volume 5 No. 2 | December 2020: 212-216
DOI: 10.15575/join. join.v5i2.612

p-ISSN: 2528-1682
e-ISSN: 2527-9165

Product Review Ranking in e-Commerce using Urgency Level Classification Approach

Hamdi Ahmad Zuhri[1], Nur Ulfa Maulidevi[2,3]
[1,2]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Indonesia
[3]PUI-PT AI-VLB (Artificial Intelligence for Vision, Natural Language Processing & Big Data Analytics), Indonesia

Article Info

Article history:

Received August 10, 2020
Revised November 03, 2020
Accepted November 27, 2020
Published December 30, 2020

Keywords:

Classification
Ranking
Text processing

ABSTRACT

Review ranking is useful to give users a better experience. Review ranking studies commonly use upvote value, which does not represent urgency, and it causes problems in prediction. In contrast, manual labeling as wide as the upvote value range provides a high bias and inconsistency. The proposed solution is to use a classification approach to rank the review where the labels are ordinal urgency class. The experiment involved shallow learning models (Logistic Regression, Naïve Bayesian, Support Vector Machine, and Random Forest), and deep learning models (LSTM and CNN). In constructing a classification model, the problem is broken down into several binary classifications that predict tendencies of urgency depending on the separation of classes. The result shows that deep learning models outperform other models in classification dan ranking evaluation. In addition, the review data used tend to contain vocabulary of certain product domains, so further research is needed on data with more diverse vocabulary.

Corresponding Author:

Hamdi Ahmad Zuhri,
School of Electrical Engineering and Informatics,
Institut Teknologi Bandung,
Jl. Ganesha No.10, Lb. Siliwangi, Kecamatan Coblong, Kota Badung, Indonesia
Email: 13516150@std.stei.itb.ac.id

1. INTRODUCTION

Product review is an important consideration for potential buyers. Consumers tend to read product reviews before paying, and product reviews significantly influence consumer decisions [1]. A good product review is expected to have several features in the form of expressions of like or dislike of the characteristics of the product (objective elements), expressions of feelings towards the product (subjective elements), or a combination of both [2]. However, along with the popularity of e-commerce, more and more reviews are circulating, and this makes it difficult for users to find informative reviews for consideration of product purchases. Therefore, product review ranking is present as a solution that makes it easier for users.

There are already several kinds of research [3][4][5] related to ranking reviews using a supervised learning approach. However, the problem is that because the supervised learning approach requires labels, experiments are conducted using the accumulation of votes as labels. The accumulated vote is the total of the upvote (user rated a review as useful one) given. However, the accumulation of votes is a cumulative process that cannot reflect the review's benefits or urgency. For example, an earlier review is considered to have more attention than a newly published review. That way, there will be a number of reviews that might be useful but have little vote because of limited exposure. Recent reviews will be labeled as "not important" and influence the results of the evaluation [6]. Problems caused by the use of accumulated votes as labels can be solved by direct human evaluation (manual labeling) of the review data. But the other problem is if the accumulation of votes is used, the label to make might be in a wide range of urgency ranks, for example, 0 to 1000 votes. However, the greater the range of urgency ratings from a review, the more difficult is the direct assessment done by humans while maintaining the consistency of assessment of each review.

The proposed solution is to try to handle those two issues (misleading vote accumulation and inconsistent manual labeling). Proposed approach is to use a classification with small range urgency ratings, for example 4 ordinal classes consisting of "not important", "less important", "important", and "very important" labels. Then determining the score will involve the probability of class selection (level of urgency) of the model so that the score remains varied even for the same class result. That way, a rating model will be obtained which, basically, is a classification model for rating review.

2. METHODOLOGY

The classification module contains two major components, namely, feature extraction and classification models. Before entering the classification module, the review data will go through a preprocessing sequence, which are expand contractions, case folding, remove numbers, remove punctuation, lemmatization, and remove stopwords.

Feature extraction used is Term Frequency times Inverse Document Frequency (TF-IDF) and word embedding, each of which has advantages and risks in extracting numeric features from text data. TF-IDF is a formal measure of how relatively small words appear in a document. The word with the highest TF-IDF value is usually a characteristic of a document [7]. Simultaneously, word embedding can be described with a case example where a word produces an embedding vector with certain properties. If a word has a meaning similar to another word, then the embedding vector for these two words will be similar [8]. TF-IDF produces simpler features so that it is more suitable for shallow learning models. One supporting reason is the fact that the TF-IDF output has a clear value. That is, the values in each column represent the same features in the review data. Therefore, this output is easily used by the shallow learning model, which, in essence, sees different data by comparing components in the same column. On the other hand, although TF-IDF eliminates sequence information in the text, this does not affect the shallow learning model's learning process. In addition, the use of TF-IDF allows N features to be selected as many features as are considered the most important so that the number of features is not too much. Also, in TF-IDF, a feature doesn't have to be a word. In TF-IDF, a feature can be two words or three words depending on the n-gram value. Whereas word embedding produces features that can be still complex, it still preserves sequence information from the text. This makes word embedding suitable for deep learning models that handle sequential input data such as LSTM and CNN. The complexity of word embedding result remains as complex as the structure of input text, but this is not a problem for the deep learning model. Unlike the shallow learning model, which is quite naive in looking at each column, the deep learning model will face a collection of input values with a collection of nodes that are interconnected so that the pattern of the input will be caught at least in certain paths even though the linkage comes from several column positions that are different. In addition, we need to limit the length of the input vector in word embedding, which results in the need for padding (usually with a value of 0) if the input is shorter or the input is truncated if the input is longer (risk of removing some information from the input).

The classification models used include shallow learning, such as Logistic Regression, Naive Bayesian, Support Vector Machine, and Random Forest, as well as deep learning, such as CNN and LSTM. Each of these algorithms has its own characteristics or assumptions. Logistic Regression is a popular and simple classification algorithm, as simple as utilizing the sigmoid function [9]. Then, Naive Bayesian is a model that was mentioned in a related paper [6]. In addition, the Naive Bayesian model uses the assumption that each feature is independent [10], which could be compatible with the case of information or urgency components that are not related in a review text. Whereas, Support Vector Machine (which aims to find the optimal hyperplane that separates classes [11]) is a model that has high popularity besides neural networks. Finally, Random Forest is a combined form of several decision trees induced from training bootstrap data samples [12], which with certain methods can provide promising results for text classification [13]. Then, Convolutional Neural Network (CNN) is a neural network that uses a special type of linear operation, namely convolution, replaces general matrix multiplication in at least one of its layers [14] and Long Short-Term Memory (LSTM) is a recurrent neural network (RNN) architecture which is specifically designed to make temporal sequence models and long-term dependencies more accurate than conventional RNN [15]. CNN and LSTM are deep learning models that handle sequential input such as text. CNN utilizes the convolution layer to simplify input while the LSTM model makes use of the LSTM layer itself to keep the data information sequentially passed.

The input of each model is the preprocessed text with its corresponding label. The class or label to do classification is in form of ordinal label (0, 1, 2, 3) which describes the level of urgency. Label 0 for "not important", label 1 for "less important", label 2 for "important", and label 3 for "very important". Label composition and data division are shown in Figure 1. How "important" a review is depends on the criteria of good review which has been explained before.

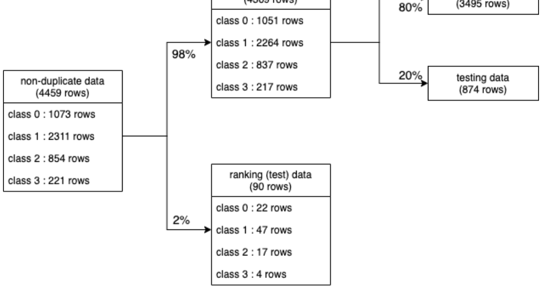

Figure 1. Data Splitting

The classification method utilizes the division of cases against these four classes into several "stages". The division of cases (into several stages) is an approach to handle multiclass labels with a binary classification approach without ignoring the order between labels. This approach (division by stage) is inspired by the one versus the rest approach in Support Vector Machine to handle multiclass problems. The one versus rest approach cannot be used because it breaks the purpose of labeling the level of urgency itself, which is for ranking. Therefore, the division of classification cases by stages is needed because it makes "separators" that still give meaning to order.

As shown in Figure 2, this separator will produce three stages, namely stage 0, which is a binary classification based on qualifications whether the review "tends to be important" (0 vs. 1, 2, 3), stage 1, which is a binary classification based on qualifications whether the review "tends to be more important" (0, 1 vs. 2, 3), and stage 2 which is a binary classification based on qualifications whether the review "tends to be very important" (0, 1, 2 vs 3). Each binary classification model that is formed will produce a probability prediction whether the review enters the positive class (has tendency) or negative (has no tendency), and all three probability values will sum to obtain an urgency score. In the end, this urgency score will be the basis of the ranking of the review data.

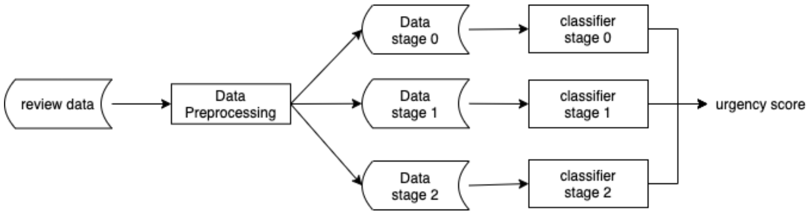

Figure 2. Ranking Architecture Based on Urgency Level

Because the reviews with label 3 are very few, the composition of positive data in "tends to be very important" case will be too small compared to the negative data. Therefore, oversampling is done by using the synthetic minority oversampling technique (SMOTE) before the learning process in the case of this stage 2 classification. SMOTE is an oversampling approach by making an example of synthesis rather than oversampling with replacement. Minority classes experience oversampling by taking each minority class sample and introducing synthetic examples along line segments connecting one or all of the nearest neighboring minority classes [16].

Every model will be used based on the defined architecture. The complete scenario of this experimental research is shown in Figure 3.

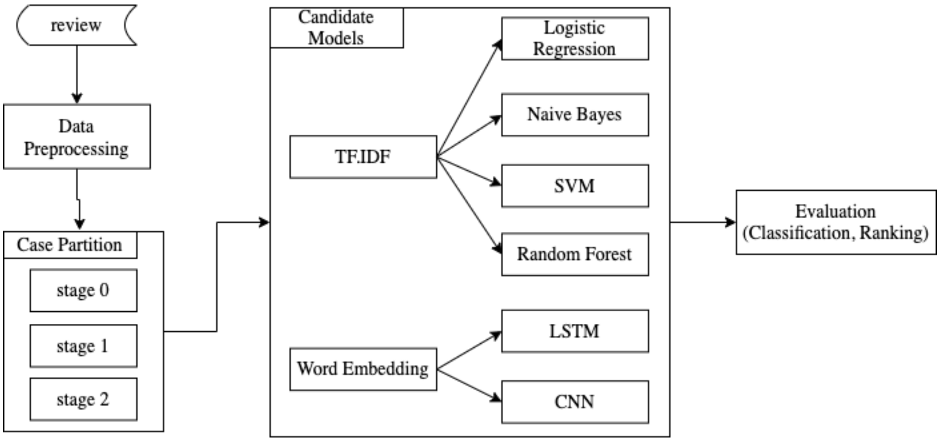

Figure 3. Experiment Scenario

3. RESULT AND DISCUSSION

There are two important kinds of evaluations to evaluate this approach. First, classification is the approach to estimate ranking problem and second, ranking as it is the end goal of this proposed solution. In the evaluation of classification performance, f-scores [17] are used. With known precision and recall values, f-scores can be calculated with the following formula.

$$F1_{score} = 2\frac{precision\ .\ recall}{precision+recall} \qquad (1)$$

The f-score value was calculated for each model's prediction process in each stage, and the results were obtained, as shown in Table 1.

Table 1. Comparison of F-scores Per Stage

Model	Stage 0	Stage 1	Stage 2
Logistic Regression	0.91	0.36	0.16
Naïve Bayes	0.87	0.07	0.19
Support Vector Machine	0.9	0.64	0.45
Random Forest	0.88	0.17	0.09
LSTM	0.94	**0.75**	**0.58**
CNN	**0.95**	**0.75**	0.54

It is found that in this case, the performance of deep learning classification (LSTM and CNN) is higher than the shallow learning classification performance (Logistic Regression, Naive Bayesian, Support Vector Machine, and Random Forest). Then in fellow shallow learning, the performance of the Support Vector Machine classification tends to be higher than the others, while Naïve Bayesian and Random Forest tend to be the lowest.

In the evaluation of ranking performance, nDCG (normalized Discounted Cumulative Gain) is used [18]. nDCG is the ratio of DCG (Discounted Cumulative Gain) to IDCG (Ideal Discounted Cumulative Gain). DCG is calculated by the following formula.

$$DCG = \sum_{i=1}^{n} \frac{2^{rel_i - 1}}{\log_2(i+1)} \qquad (2)$$

It is noted that *i* is the ranking position of a particular review, and *n* is the number of reviews in the ranking group. While the *rel* is the weight of the review at position *i*. IDCG is calculated in the same way, but position *i* is determined based on ground truth. The nDCG scores are shown in Table 2.

Table 2. Comparison of nDCG values

Model	All	Top 1	Top 3	Top 5	Top 10	Top 20	Top 30	Top 45
Logistic Regression	0.9727	0.9555	0.9552	0.9496	0.9587	0.9267	0.9157	0.9173
Naïve Bayes	0.9134	0.7444	0.789	0.7506	0.6785	0.6965	0.7502	0.8137
Support Vector Machine	0.9715	0.9888	0.951	0.9641	0.9175	0.9194	0.9272	0.9191
Random Forest	0.9751	0.9555	0.9618	0.9551	**0.9616**	0.9181	0.9285	0.9249
LSTM	**0.984**	**1**	**0.9973**	0.9595	0.9468	**0.9455**	0.9532	0.9413
CNN	0.983	**1**	0.9926	**0.968**	0.9545	0.9386	**0.9633**	**0.9435**

It is found that in this case, the ranking performance utilizing deep learning (to be exact, deep learning models that handle sequences in data, namely LSTM and CNN) tends to be higher than the ranking performance utilizing shallow learning (Logistic Regression, Naive Bayesian, Support Vector Machine, and Random Forest). These results are in line with the results of the classification performance evaluation discussed earlier.

Then, in fellow shallow learning, Naive Bayesian ranking performance is very much lower than other models. This indicates that the assumption of independent features cannot be applied in this problem.

In addition, the performance of the Logistic Regression, Support Vector Machine, and Random Forest ranking is quite competitive with values that are slightly below the performance of LSTM and CNN, even in some cases, these models are better than LSTM or CNN. This is not so in line with the results of the classification evaluation obtained.

To sum up, in solving the problem of using upvote as a label, the proposed approach can give good performance in ranking depending on the model used, but the result also shows that f-score and nDCG score of models are not correlated enough to represent or estimate score each other.

4. CONCLUSION

This experiment shows how various models behave in classification and ranking based on 4 ordinal labels of urgency. The review ranking model with the best performance in the classification evaluation and ranking evaluation, obtained from experiments, uses deep learning, namely LSTM and CNN. Based on the evaluation results, it appears that the results of the classification evaluation of a model are not necessarily aligned or positively correlated with the results of the ranking evaluation so that the determination of the best review ranking model cannot be based on classification evaluation but a ranking evaluation.

For the development of this solution in the future, it is better to use more and varied review data so that experiments can be conducted by considering various vocabularies from various product domains, such as food, clothing, and digital products. The experiment aims to ascertain whether a ranking system with a variety of vocabulary is still feasible to apply. It is also better if the labeling, especially the ranking label, is done by a number of people and is studied in depth (although it takes a long time) so that the bias or inconsistencies in the test data or ground truth used can be minimized.

ACKNOWLEDGEMENTS

We thank our colleagues from School of Electrical Engineering and Informatics Institut Teknologi Bandung who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper.

5. REFERENCES

[1] P.-Y. Chen, S. Wu, and J. Yoon, “The impact of online recommendations and consumer feedback on sales,” *ICIS 2004 Proc.*, p. 58, 2004.

[2] A. Ghose and P. G. Ipeirotis, “Designing novel review ranking systems: predicting the usefulness and impact of reviews,” in *Proceedings of the ninth international conference on Electronic commerce*, 2007, pp. 303–310.

[3] H.-Y. Hsieh and S.-H. Wu, “Ranking online customer reviews with the SVR model,” in *2015 IEEE international conference on information reuse and integration*, 2015, pp. 550–555.

[4] S. Saumya, J. P. Singh, A. M. Baabdullah, N. P. Rana, and Y. K. Dwivedi, “Ranking online consumer reviews,” *Electron. Commer. Res. Appl.*, vol. 29, pp. 78–89, 2018.

[5] J. P. Singh, S. Irani, N. P. Rana, Y. K. Dwivedi, S. Saumya, and P. K. Roy, “Predicting the ‘helpfulness’ of online consumer reviews,” *J. Bus. Res.*, vol. 70, pp. 346–355, 2017.

[6] J. He, K. Niu, Z. He, S. Wang, and Z. Bie, “A supervised method for ranking reviews based on latent structure features,” in *2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA)*, 2016, pp. 88–92.

[7] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng, “Parsing with compositional vector grammars,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2013, pp. 455–465.

[8] A. Burkov, *The Hundred-Page Machine Learning Book, Quebec City, Canada, 2019, ISBN 978-1999579517*. Taylor & Francis, 2020.

[9] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic regression*. Springer, 2002.

[10] I. Rish, “An empirical study of the naive Bayes classifier,” in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, vol. 3, no. 22, pp. 41–46.

[11] I. El-Naqa, Y. Yang, M. N. Wernick, N. P. Galatsanos, and R. M. Nishikawa, “A support vector machine approach for detection of microcalcifications,” *IEEE Trans. Med. Imaging*, vol. 21, no. 12, pp. 1552–1563, 2002.

[12] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[13] A. Onan, S. Korukoğlu, and H. Bulut, “Ensemble of keyword extraction methods and classifiers in text classification,” *Expert Syst. Appl.*, vol. 57, pp. 232–247, 2016.

[14] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1, no. 2. MIT press Cambridge, 2016.

[15] H. Sak, A. W. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” 2014.

[16] O. Chapelle, P. Shivaswamy, S. Vadrevu, K. Weinberger, Y. Zhang, and B. Tseng, “Boosted multi-task learning,” *Mach. Learn.*, vol. 85, no. 1–2, pp. 149–173, 2011.

[17] D. L. Olson and D. Delen, *Advanced data mining techniques*. Springer Science & Business Media, 2008.

[18] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of IR techniques,” *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.