Review Rating Prediction: An investigation of the performance of different Word Embeddings and Classification Models

1st Dao Thi Nguyen
StudentID: 20212316
DC836.MSc Computing.Data Analytics
dao.nguyen7@mail.dcu.ie

2nd Chau Minh Nguyen
StudentID: 20212456
DC836.MSc Computing.Data Analytics
chau.nguyenminh4@mail.dcu.ie

***Abstract*—Online review references play an important role in consumers' purchasing decision. This is because these reviews can be informative regarding perspective and experience about the quality of products or services. The rating stars on reviews are expected to be a valuable factor for consumers' purchasing decision. However, the quantitative rating mechanism can be missing in certain online systems such as social media networks. Therefore, in this paper, we conducted an investigation and experiments with respect to the existing embedding approaches in Natural Language Processing as well as Machine Learning and Deep Learning algorithms in the review rating prediction problem. The research has been implemented based on a subset of the Yelp review dataset. A range of predicting techniques, i.e. Random Forest, Logistic Regression, Decision Tree and a Recurrent Neural Network (RNN) based model are utilized along with several embedding techniques for feature transformation to predict the review rating. The results have shown promising performance in the accuracy of the RNN-based and Mean Word2Vec model, which outperforms other embedding and predicting approaches.**

***Index Terms*—Review Rating prediction, Classification techniques, Yelp dataset, Recurrent Neural Networks, Machine Learning**

I. Introduction

Online text-based informative reviews are of great significance in a consumer's decision-making processes in using a service [1]. Typically, a quantitatively rating mechanism can be used to represent users' satisfaction with the service, e.g. after using the service, users can subjectively give stars on the scale from 1 to 5. The higher number of stars a user gives, the greater level of happiness about the service he or she conveys. The overall rating can serve as a reference for other customers to make wiser decision when they are considering using the service. Nevertheless, several online review websites merely allow online text reviews without an associated rating mechanism. In other words, users are unable to rate the services or products in a quantitative manner. For instance, the websites may integrate their review section with the comment features of popular social networks such as Facebook or Twitter, causing that the users' opinions are merely represented by text. As a consequence, it can be challenging for a new user to research the general perspective of the community about the services. Therefore, it would be essential to explore the development of an automated review rating prediction system to predict the rating of unrated customer reviews. In this paper, we aim to present state-of-the-art approaches in Natural Language Processing (NLP) in the predictions of quantitative rating for "text-only" reviews. In general, the target of this article is to answer the following research question:

- Can the quantitative rating of reviews be predicted by using their text content? If yes, which existing approaches deliver the best performance in terms of prediction accuracy?

The research has been carried out based on the Yelp dataset which contains a huge volume of reviews in either text or associating starred rating. In addition, various NLP techniques, i.e. *Word2Vec*, *Doc2Vec, TF-IDF* embeddings have been used to transform the text data into numeric vectors. Then, a range of machine learning algorithms is utilised to predict the starred rating of the reviews. A comparison of different results from different approaches has also been discussed.

The rest of this paper is structured as follows: Section 2 provides the literature review on review rating predictions. Section 3 presents the research method of the paper. The experimental results are presented and discussed in Section 4, followed by a conclusion and future work in Section 5.

II. Related work

A. Embedding techniques

Performing analysis on text data requires a transformation of text into numerical representation so that analytical techniques and algorithms can capture meanings and relationship of texts. In *NLP*, word embedding has been shown to be the dominant approach to resolve this problem [2] by representing an individual word as a vector in a predefined vector space. This process is called *embedding* [2]. Several research efforts are presenting different contributions to the improvement of the word embedding process. The term *Frequency-Inverse Document Frequency* (TF-IDF) [3] is a statistical-based measurement for weighting the value of words by the frequency of the appearance of a word in a document and the number of documents in the dataset that contain that single word.

However, the *TF-IDF* approach skips the order of words in sentences, which leave out the semantic meaning, and a sparse matrix could create resource intensive. To overcome the issues of semantic and resource-consuming, *Word2Vec* [4] was introduced as an efficient method for learning high-quality vector representations of words using neural networks. While *Word2Vec* model generates vectors for each word in a document, *Doc2Vec* [5] embeds each document of the data set into a vector space. Our project absorbed these *word embedding* methods and selectively apply for the data transformation process.

B. Review rating prediction techniques

Review rating prediction can be considered as a classification problem [6]–[8]. Two types of classification approaches can be found in the context of predicting review rating. The first approach is to manually extract the features which are the input data for machine learning-based classification algorithms. Popular *machine learning-based* models employed in sentiment analysis are *Logistic Regression* [9], *Decision Tree* [10] and *Random Forest* [11]. In general, these models gained success with respect to the feature engineering work. The success of these standard machine learning-based methods generally regarding the feature engineering work such as the bag-of-words (BOW) representation and manual sentiment lexicon.

The second approach is to employ a *Deep Neural Network-based* architecture, i.e. Recurrent Neural Networks (RNN). From this architecture, salient and important features in the given text can be learned automatically, which facilitate achieving promising accuracy in classifying the given text. RNN has a variety of variants with different architectures and mapping functions such as *Long Short-Term memory Network* (LSTM) [12]. LSTM was proposed to especially solve hard long term lag issues. In 2014 [13], the authors proposed a *Gated recurrent unit* (GRU) based RNN, which is an improved version of the LSTM. The structure of GRU gives it the ability to ignore the irrelevant data using the reset gates and using update gates to capture the contextual data from feature vectors, resulting in high accuracy in sentiment classification. *Bidirectional-GRU* (Bi-GRU), an extension of GRU, adds another layer to the GRU networks allowing the model to exploit the temporal information flow in both directions. Similarly, *Bidirectional-LSTM (Bi-LSTM)* is proposed in [14].

In 2019, ensemble learning using several architectures such as Bi-LSTM, Bi-GRU and CNN is proposed in [15]. These models gained great success in solving the sentiment classification problem. The success of these methods depends on its ability to learn text representation from the data without the fatigue of performing feature engineering processes, and the ability to define the semantic relationships between context words in a more extensible way than standard machine learning-based methods.

In 2020, the authors in [6] introduced a framework to predict rating stars for customer review with review sentiment. The framework based on Bi-GRU model architecture, and the embedded features considering in the review content. This approach has demonstrated a more reliable rating prediction [6].

The approaches discussed above have demonstrated their effectiveness in review classification problems. In this paper, we seek to use traditional machine learning-based methods (*Decision Tree*, *Logistic Regression* and *Random Forest*) and *RNN-based* methods (*ensemble Bi-LSTM* and *Bi-GRU* models together) on different feature representations extracted by different embedding techniques. The embedded data have been used to train the models for predicting reliable rating stars for unrated reviews.

III. Methodology

A. Data exploration

The dataset used for the research was acquired from Yelp's businesses and reviews data from 2017- 2019, which contained 2.8 million customers reviews for 200,000 businesses. In the scope of this project, we focused on the reviews of businesses in the restaurant category only. The investigation of the dataset includes the study of the review rating distribution, text length distribution and the contribution of useful review on the full set of data. Based on the result, we made a selection for a data subset that seems to be the most representative of the full set for further analysis. After pre-processing the data subset, we examine the distribution of top unigrams before and after removing stop words and calculate the sentiment polarity score of each version.

It is crucial to understand the data before any analysis is conducted. Originally introduced by John Tukey in 1970, Exploratory Data Analysis (EDA) is a widely-used method to obtain a better understanding of data set variables and the relationships between them [16]. Following the instruction of [16], we applied several EDA techniques to test our hypothesis, check assumptions and choose a representative subset of the data set. The histogram of review rating distribution, (Figure 1a), revealed that the 5-star review, with 1,589,266 records (41.14% of total reviews), is the biggest contributor and the 2-star reviews are the least with 37.023 records (9.02% of total reviews). From the review text length distribution, (Figure 1b), and bar chart of useful, (Figure 1c), it can be seen that 100-199 words and the reviews, that were not marked useful by other users, were by far the highest proportions of reviews. Since the label features being used in our prediction are the review rating, we took the majority of groups and concentrate on the similarity of the shape of the diagram to form a subset of data (Figure 2).

To gain better attention for the important words, a comparison between the highest frequent unigrams (single word) before and after lemmatizing and removing stop words has been made. Figure 3 illustrates the top 20 word counts from the dataset before and after pre-processing. It can be seen that before removing stop words, most of the top list comprises of less meaningful words such as articles (e.g. *a, an, the*) and preposition (e.g. *to, of, in* etc.). On the other hand, after the pre-processing phase, the top list has been able to show the

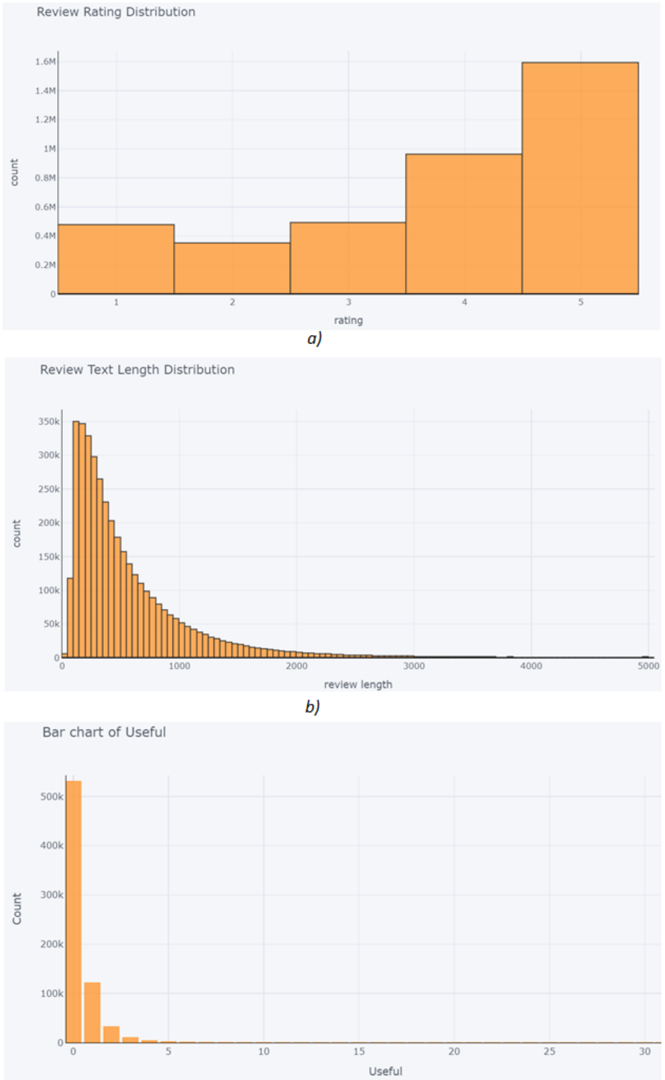

Fig. 1. Distributions in full data set.

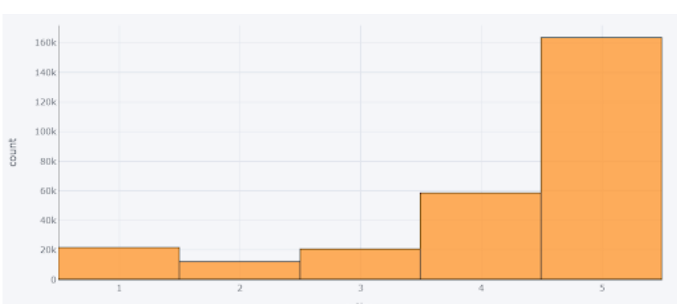

Fig. 2. Review Rating Distribution after preprocessing.

words with better semantic richness such as *food, good, love, amazing* etc.

B. KDD process

In this paper, the KDD process is utilised to predict rating stars for text reviews in the Yelp reviews dataset. In particular, this approach consists of the following steps:

1) Data Selection: Based on the result of data exploration mentioned above, the sample subset is selected by filtering the originally acquired dataset with three criteria including business_category = "Restaurant", useful = 0, and review text length in [100,199]. The selected sample subset contains 275,197 restaurant reviews with a 1-5-scaled starred rating.

2) Data pre-processing: The sample data will be cleaned and pre-processed by removing punctuation, special characters, digits and stop words. This step can help to reduce the size of data and remove the meaningless characters/words. Additionally, all words have been converted to lowercases. The lemmatization will also be applied to convert the words into their root words, reducing the number of unique words in the dataset.

3) Transformation: In the next stage, cleaned data will be transformed by different techniques consisting of *bag of word*, *Word2Vec*, *Doc2Vec*, *TF-IDF*, *word embeddings*, and *Paragraph Vector-Distributed Memory* (PV-DM) model.

4) Model and Training: Finally, the different kinds of transformed data will be utilized to train models consisting of *Decision Tree*, *Logistic Regression*, *Random Forest* and *Recurrent Neural Network* based model (*Bi-LSTM, Bi-GRU*) to predict ratings for reviews.

5) Evaluation: We adopt the popular evaluation metrics namely *Accuracy, Precision, Recall* and *F1-score*, which have been used in several previous papers [6], [17]. These metrics have been utilized to evaluate and compare the performance of the trained classification models.

- The *Accuracy* can be considered as the most popular evaluation metric for binary or multi-class classification problems [18]. In this metric, the quality of the rating prediction or classification is evaluated by calculating the percentage of right classifications over the total number of evaluated instances. The *Accuracy* metric is not only applicable for multi-class problems, but it is also easy to calculate, score and comprehend.
- *Precision* represents the percentage of predicted positive reviews is positive.
- *Recall* indicates the percentage of positive reviews that classified accurately.
- *F1-score* (F Score or F Measure) conveys the balance between the precision and the recall [18]. It can be used to compare two models with low accuracy and high recall which is normally complicated.

IV. Experimental Setup

A. Word Embeddings

We test the efficient of *word2vec* and *doc2vec* by creating a vector for each document then applied on a basic *Logistic*

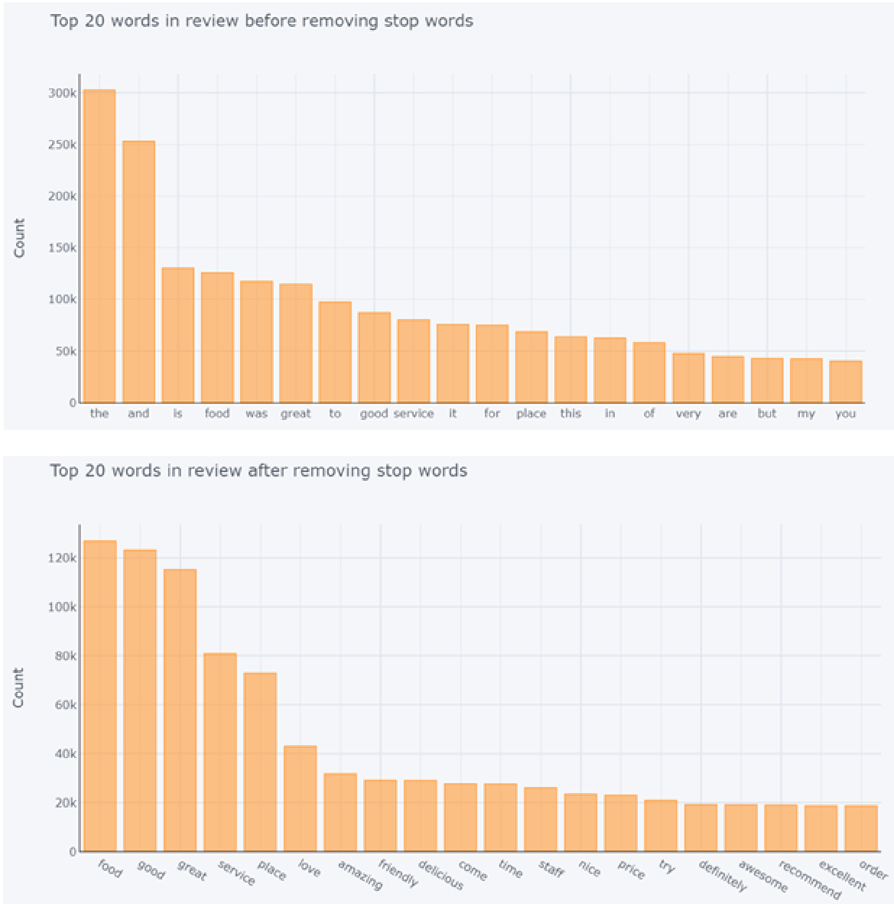

Fig. 3. Top 20 word counts before/ after removing stopw words.

Regression using the *sklearn* library with default configuration. For the *word2vec*, to perform its word vector into one vector for a document, we consider several transformations as follows:

- *Mean Word2Vec*: compute average all word vector occurred in a single text;
- *TF-IDF weighted Word2Vec*: adopt TF-IDF as weights for each word embedding before taking average;
- *Mean GloVe*: average pre-trained GloVe word vectors from a text.

For doc2vec, we directly train doc vectors using the *Paragraph Vector - Distributed Memory* model [5]. We also concatenate the *word2vec* and *doc2vec* vectors to verify if having more feature could elevate the performance. The source code was inherited from [19].

B. Machine learning approaches

For the investigation, we apply three supervised machine learning algorithm: *Logistic Regression*, *Decision Tree* and *Random Forest* following default configuration of *scikit-learn* library [20]–[22]. The calculation show that *Logistic Regression* offer the best results among three for every embedding models.

C. Recurrent Neural Network model

Bi-LSTM and *Bi-GRU* are implemented with the support of *TensorFlow* and *Keras* libraries in Python for training rating prediction models. The features are selected based on the data pre-processing mentioned above.

For building and training the model, the review texts have been tokenized and converted to sequences. Each review document has been limited to 200 words. While the long documents which are greater than 200 words are truncated, the shorter ones are padded with zeros.

Two kinds of word embedding approaches are employed to do experiments on the model, described below.

- *Pre-trained GloVe* word embeddings [23] which has been trained on a huge dataset of 27 billion tokens (Twitter tweets) with a vocabulary of 1.2 million words and embedding vector size of 200 dimensions. Both Twitter and reviews are considered to include non-standard English and unusual punctuation. Therefore, Twitter vocabulary seemed fitting with Yelp reviews.
- *Self-trained Word2Vec* embeddings which are trained on a subset of the Yelp review dataset mentioned in Section 3.

The model experimented within this study is the RNN-based

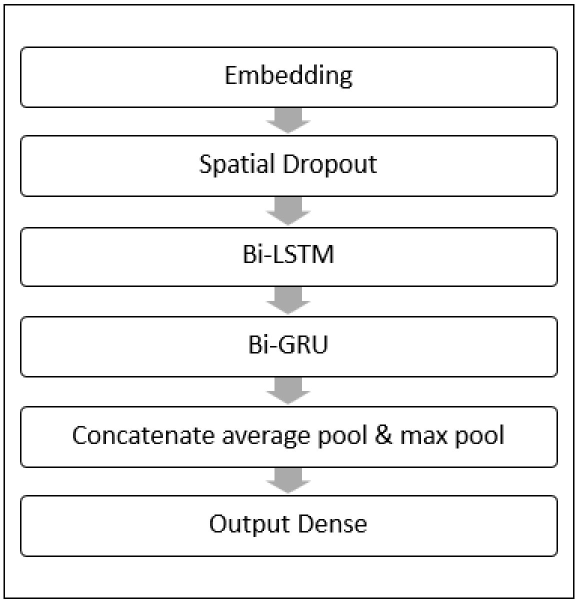

Fig. 4. RNN-based model architecture used in this paper.

model (Bi-GRU, Bi-LSTM ensembled together) [24]. As can be seen in Figure 4, the model architecture is described as follows:

- *Word Embedding layer* is used for mapping from integer indices (words) to their 200-dimensional dense vector (their embeddings), expanding each token to a more massive vector. A word is, therefore, represented in a meaningful way.
- *Drop out layer* is employed for handling over-fitting and make better generalizations during the training process. The 1D Spatial dropout is to drop an entire row (one-dimensional vector) out of the embedded matrix with a 0.5 dropout rate. In other words, this type of dropout drops words from the sequence.
- *Bi-LSTM layer* takes as input the word embeddings as a matrix consisting of the vertically stacked embedding vectors corresponding to the words included in the review. This matrix can be thought of as a sequence of embedded words. Each of these embedding vectors is fed to the bidirectional LSTM at their respective time step and the final time step. Hence, this layer is constructed based on LSTM.
- *Bi-GRU layer* is constructed based on GRU. It takes as input the output of the Bi-LSTM layer with the lower number of parameters (from 81648 in Bi-LSTM to 32256)
- *Concatenation layer* includes *Global max pooling* and *Global average pooling*. This layer is then applied to the output of the Bi-GRU layer and the resulting vectors are concatenated to form a one-dimensional array.
- *Output dense layer* is a regular neural network layer. The output of this layer going through the softmax function to give the prediction.

The first and second hidden layer will have 40 memory units and the output layer will be a fully connected layer that outputs one value per timestep. A softmax activation function which often used for multi-class classification is utilized to guarantees a well-behaved probability distribution function. Because this is a multi-class classification problem, the multiclass log loss (multiclass-cross entropy in Keras) is used. The efficient ADAM optimization algorithm is also employed to find the weights and the accuracy metric is calculated and reported each epoch. The dataset is splitted into training (80%) and test (20%) set. The model is trained in 10 epochs and validated using test data entries.

V. Experimental Results

In this section, we discuss the performance of different word embedding models on text classification models. The source code of the experments in this paper can be found in [25].

TABLE I
CLASSIFICATION PERFORMANCE OVER DIFFERENT WORD EMBEDDING MODELS USING LOGISTIC REGRESSION

***Using Logistic Regression**	***Accuracy***	***F1 score***	***Precision***	***Recall***
Mean Word2Vec	0.693	0.651	0.65	0.69
TF-IDF weighted Word2Vec	0.686	0.640	0.64	0.69
PV-DM Doc2Vec	0.663	0.608	0.61	0.66
TF-IDF Word2Vec + Doc2Vec	0.69	0.651	0.65	0.69
Mean GloVe	0.680	0.680	0.63	0.68

TABLE II
PERFORMANCE OVER DIFFERENT MACHINE LEARNING APPROACHES IN THE ACCURACY METRIC

	Logistic Regression	***Decision Tree***	***Random forest***
Mean Word2Vec	0.693	0.545	0.668
TF-IDF weighted Word2Vec	0.686	0.532	0.663
PV-DM Doc2Vec	0.663	0.455	0.61
TF-IDF Word2Vec + Doc2Vec	0.69	0.53	0.66
Mean GloVe	0.67	0.51	0.64

A. Embedding models Performances

Table 1 summarized the performance over different word embedding while putting in logistic regression. Overall, the *Doc2Vec* model performed the worst with 0.663 in accuracy while other *Word2Vec-based* methods' results are all above 0.69. Furthermore,for *Word2Vec*, concatenation present

TABLE III
PERFORMANCE OF DIFFERENT PREDICTING MODELS OVER DIFFERENT DATA TRANSFORMATION APPROACHES

Model Name	Feature transformation	Self-train word2vec				GloVe pre-trained embeddings			
		Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
Random Forest	concat tfidf and doc2vec	0.66	0.59	0.66	0.58	0.63	0.56	0.63	0.52
	tfidf	0.663	0.60	0.66	0.59	0.63	0.55	0.63	0.52
	mean word2vec	**0.668**	0.60	0.67	0.60	0.64	0.57	0.64	0.54
Logistic Regression	concat tfidf and doc2vec	0.69	0.65	0.69	0.65	0.68	0.63	0.68	0.64
	tfidf	0.686	0.64	0.69	0.64	0.66	0.60	0.66	0.60
	mean word2vec	**0.693**	0.65	0.69	0.65	0.67	0.62	0.67	0.62
Decision Tree	concat tfidf and doc2vec	0.53	0.54	0.53	0.54	0.49	0.50	0.49	0.50
	tfidf	0.532	0.54	0.53	0.54	0.49	0.50	0.49	0.49
	mean word2vec	**0.545**	0.55	0.55	0.55	0.51	0.51	0.51	0.51
RNN (Bi-GRU, Bi-LSTM)		**0.733**	0.71	0.73	0.71	**0.736**	0.71	0.74	0.72

a slightly improvement (0.692) while considering its participants outcome in individual run (0.686 and 0.692). However, this result is not significantly different comparing to *Mean self-trained Word2Vec*. In other word, using both *Word2Vec* and *Doc2vec* appears to be unnecessary. *Mean self-trained Word2Vec* has shown the best performance (0.693) even though it was the most straightforward method, better than pre-trained GloVe (0.68 in accuracy).

B. Predicting models Performance

Table II shows the experimental results of rating star prediction on different models with different types of *Word2Vec* embedings. It can be seen that the performance of word embedding-based models have been consistent throughout traditional machine learning algorithms. For *self-trained vectors*, the *Mean Word2Vec* models present the highest accuracy and there are no significant effection while combining it with other methods including *TF-IDF weighted* and concatenate with *Doc2Vec* models.

In Table III, the result has shown that *RNN (Bi-GRU, Bi-LSTM)* model has achieved the outstanding performance regarding *Accuracy, Precision, Recall, and F1-score* with both input types of Word2Vec embeddings, i.e. self-trained Word2Vec and pre-trained GloVe, in comparison with other models. Furthermore, in terms of the RNN-based model, it obtains a light more accuracies (0.003) by using pre-trained word embeddings (GloVe) than using *self-trained word2vec*, at 0.736 and 0.733 respectively.

On the other hand, in terms of the standard machine learning-based models, *self-trained word2vec embeddings* models have been shown to achieve higher performance in all evaluated metrics than the Glove-based models.

VI. CONCLUSIONS AND FUTURE WORK

Online text-based informative reviews play an integral part in customers' consuming decision-making. Nevertheless, collecting useful information from a huge number of reviews is always challenging for consumers. As a result, predicting star ratings for online reviews is necessary. In this paper, we conducted researches about solution for this issue. Different Word Embedding methods are experienced on Decision Tree, Random Forest, Logistic Regression and RNN-based (Bi-GRU, Bi-LSTM) models are selected and experimented on a subset of Yelp dataset with different feature representations. The experimental results appear to show the following findings in the multi-class classification task using the Yelp review dataset:

- The *RNN-based* model has been shown to achieve the best performance regarding *Accuracy, Recall, Precision* and *F1-score* in the classification task in overall.
- The *Logistic Regression* model has been standing out in predicting the review ratings in comparison with other standard *machine learning-based* classifiers.
- The *Glove embedding approach* brings better performance for the *RNN-based* model in predicting review rating stars, while the *self-trained Word2Vec* approach appears to be a better choice for *machine learning-based* models in predicting ratings.
- The *Mean Word2Vec* has been shown to be outperformed other *Word2Vec-based* and *Doc2Vec* methods.

In future work, we would consider more feature transformation techniques to analyse the text review. Moreover, it would be interesting to conduct more experiments on these mentioned models with other datasets in other languages such as Vietnamese.

REFERENCES

[1] F. Li, N. N. Liu, H. Jin, K. Zhao, Q. Yang, and X. Zhu, "Incorporating reviewer and product information for review rating prediction," in *Twenty-second international joint conference on artificial intelligence*, 2011.

[2] Y. Li and T. Yang, "Word embedding for understanding natural language: a survey," in *Guide to big data applications*. Springer, 2018, pp. 83–104.

[3] B. Trstenjak, S. Mikac, and D. Donko, "Knn with tf-idf based framework for text categorization," *Procedia Engineering*, vol. 69, pp. 1356–1364, 2014.

[4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[5] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*. PMLR, 2014, pp. 1188–1196.

[6] B. H. Ahmed and A. S. Ghabayen, “Review rating prediction framework using deep learning,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–10, 2020.

[7] L. PaNgB, “Exploiting class relationships for sentiment categorization with respect to rating scales,” *IN: ProceedingsofACL r05*, 2005.

[8] C. S. C. Reddy, K. U. Kumar, J. D. Keshav, B. R. Prasad, and S. Agarwal, “Prediction of star ratings from online reviews,” in *TENCON 2017-2017 IEEE Region 10 Conference*. IEEE, 2017, pp. 1857–1861.

[9] A. Tyagi and N. Sharma, “Sentiment analysis using logistic regression and effective word score heuristic,” *International Journal of Engineering and Technology (UAE)*, vol. 7, pp. 20–23, 2018.

[10] A. Bayhaqy, S. Sfenrianto, K. Nainggolan, and E. R. Kaburuan, “Sentiment analysis about e-commerce from tweets using decision tree, k-nearest neighbor, and naïve bayes,” in *2018 international conference on orange technologies (ICOT)*. IEEE, 2018, pp. 1–6.

[11] P. Karthika, R. Murugeswari, and R. Manoranjithem, “Sentiment analysis of social media network using random forest algorithm,” in *2019 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*. IEEE, 2019, pp. 1–5.

[12] S. Hochreiter and J. Schmidhuber, “Lstm can solve hard long time lag problems,” *Advances in neural information processing systems*, pp. 473–479, 1997.

[13] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[14] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, and N. A. Smith, “Transition-based dependency parsing with stack long short-term memory,” *arXiv preprint arXiv:1505.08075*, 2015.

[15] Z. Rebiai, S. Andersen, A. Debrenne, and V. Lafargue, “Scia at semeval-2019 task 3: sentiment analysis in textual conversations using deep learning,” in *Proceedings of the 13th international workshop on semantic evaluation*, 2019, pp. 297–301.

[16] S. Mukhiya and U. Amed, *Hands-On Exploratory Data Analysis with Python*. Packt Publishing, 2020, vol. 1.

[17] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[18] M. Hossin and M. Sulaiman, “A review on evaluation metrics for data classification evaluations,” *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, p. 1, 2015.

[19] T. Lin. (2021, Mar.) Performance of different word embeddings on text classification. [Online]. Available: https://towardsdatascience.com/nlp-performance-of-different-word-embeddings-on-text-classification-de648c6262b

[20] (2021) Logistic regression. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[21] (2021) Decision tree classifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[22] (2021) Random forest classifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[23] S. team. (2018) Pre-trained glove word embedding. [Online]. Available: http://nlp.stanford.edu/data/glove.twitter.27B.zip

[24] K. authors. (2018, Jan.) Predicting yelp ratings from review text. [Online]. Available: https://www.kaggle.com/athoul01/predicting-yelp-ratings-from-review-text

[25] (2021) Source code. [Online]. Available: https://github.com/dao-nguyen0912/CA683_Assignment_star_rating_prediction

VII. Declaration on Plagiarism

We declare that this material, which we now submit for assessment, is entirely my own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of my work. we understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should we engage in plagiarism, collusion or copying. We have read and understood the Assignment Regulations. We have identified and included the source of all facts, ideas, opinions, and viewpoints of others in the assignment references. Direct quotations from books, journal articles, internet sources, module text, or any other source whatsoever are acknowledged and the sources cited are identified in the assignment references. This assignment, or any part of it, has not been previously submitted by me or any other person for assessment on this or any other course of study.

We have read and understood the referencing guidelines found at http://www.dcu.ie/info/regulations/plagiarism.shtml, https://www4.dcu.ie/students/az/plagiarism and/or recommended in the assignment guidelines.

Name: Dao Thi Nguyen and Chau Minh Nguyen

Date: 17/04/2021