The 9th International Conference on Web Research
ICWR2023
May 3th-4th, 2023, Tehran, Iran

Customer Loyalty Prediction of E-marketplaces Via Review Analysis

Fakhroddin Noorbehbahani*, Soroush Bajoghli, Hooman Hoghooghi Esfahani
Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran
noorbehbahani@eng.ui.ac.ir, s.bajoghli@eng.ui.ac.ir, h.hoghooghi@eng.ui.ac.ir

***Abstract*— Today, leveraging analytical CRM to maximize values for both customers and businesses is one the most important critical success factors Predicting customer loyalty enables businesses to differentiate among customers for conducting relationship marketing and implementing effective customer extension tactics. In this paper, we analyze customers' reviews on the Digikala e-marketplace to predict their loyalty. We employ NLP, deep learning, and conventional machine learning methods and evaluate the results to find the best prediction model. Two experiments are conducted to evaluate the results: binary and 3-class loyalty prediction. In the binary setting, the Random Forest and Naïve Bayes algorithms outperformed the other tested classification methods and achieved an accuracy of 89%. In the 3-class setting, the Random Forest classification method achieved the best performance among all other machine learning algorithms with an accuracy of 67%. The evaluation results imply that businesses could benefit from using the Random Forest classification algorithm to predict customer loyalty through review analysis successfully.**

Keywords— Customer Loyalty; Natural Language Processing; Machine Learning; Deep Learning

I. INTRODUCTION

Nowadays, analytical CRM has become a crucial aspect of any customer-led business enabling organizations and companies to segment and model customers, gauge business performance, and deliver great value to customers. Customer loyalty refers to a customer's willingness to continue doing business with a particular company or brand. Considerable research has been conducted on customer loyalty, with two major approaches used to define and measure it. One approach is based on behavior, while the other is based on attitude [1].

The practicality of the behavioral definition of loyalty lies in the fact that actions, rather than attitudes, drive sales and profits. However, although it requires effort, examining the reasons behind weak or negative customer attitudes can assist companies in identifying obstacles to purchasing. On the other hand, recognizing strong or positive attitudes can enable companies to comprehend the factors underlying competitor-resistant commitment [1].

Customers who exhibit loyalty to a brand by returning repeatedly are the most valuable. These loyal customers not only make repeat purchases themselves but also serve as brand ambassadors by recommending a brand to others. However, measuring this loyalty might be a challenge, despite its significance. Acquiring this knowledge can aid in evaluating the effectiveness of a brand's long-term customer retention strategies

Reichheld introduced NPS (Net Promoter Score) in 2003 through a Harvard Business Review article titled “The Only Number You Need to Grow” [2]. This method assesses customer loyalty towards a company's products or services post-use and involves a single-question survey, where customers are asked to rate their likelihood of recommending the company to a friend or colleague [3]. According to Reichheld’s study [2], conducting extensive surveys is unnecessary as NPS is a dependable standalone approach to measure customer loyalty.

The development of technology and the widespread use of the Internet have opened up new possibilities for online stores [4]. One of the most crucial aspects of any business is customer feedback, as it helps marketers and entrepreneurs enhance the customer experience and attract new customers.

Digikala is the largest online store/marketplace in Iran, catering to a diverse range of products for all age groups and segments of society. Their e-marketplace features digital goods, home appliances, beauty and health, culture and art, sports, and entertainment [5]. Digikala has emerged as the most popular and frequently visited website in Iran, capturing over 85% of the online retail market [6].

On the Digikala website, customers can provide reviews on specific products and even recommend them for purchase. The option to recommend a product is available when customers write reviews, and they can choose to either recommend the product, not recommend it, or leave no opinion about it. NPS only measure customer loyalty and couldn’t predict loyalty. Customer loyalty prediction enables businesses to differentiate among customers for conducting relationship marketing and implementing customer extension tactics effectively.

This study evaluates various machine learning and deep learning algorithms for analyzing Persian reviews to predict customers’ attitudinal loyalty to the Digikala e-marketplace. The research questions are as follows:

RQ1: What is the best conventional machine learning or deep learning technique for predicting customer loyalty based on review analysis in Persian?

RQ2: Does customer loyalty prediction benefit from review analysis using machine learning techniques?

The rest of this paper is organized as follows. Section II presents related work in this research area. Section III describes the methodology for customer loyalty prediction. Section IV outlines the results and discussion, and Section V offers the main conclusion.

II. RELATED WORK

Our investigation reveals no prior studies that specifically aim to predict customer loyalty through review analysis in Persian. This section provides a review of the techniques used for sentiment and satisfaction analysis in the Persian language using reviews, as it shares similarities with predicting customer loyalty based on reviews.

The goal of sentiment analysis is to identify individuals' emotions and viewpoints from their online reviews. It is commonly used in the business world to analyze social data, evaluate brand perception, and comprehend customer sentiments. While numerous studies have focused on sentiment analysis for the English language, resources for the Persian language are limited [7].

Dashtipour and his colleagues developed a hybrid Persian sentiment analysis framework and applied the integrating dependency grammar-based rules and deep neural networks to optimize polarity detection [8]. This study focuses on new rules based on the dependency for the analysis of Persian emotions, hierarchical relationship between keywords, words, word order, polarities (positive and negative reviews) in long sentences, and constraints based on the frequency of synchronous words. The method used in this study has a higher accuracy of 18-16% compared to traditional machine learning methods and 6-9% compared to deep learning methods.

Dehkharghani et al. introduced a new method for Persian sentiment analysis that utilizes discourse and external semantic information. The approach presented in their paper consists of two parts: the first method employs a classifier combination, while the second approach uses deep neural networks. Both methods incorporate data from the local discourse and external knowledge base. Additionally, various linguistic challenges, including negation and emphasis, manifest at different levels, such as the word, sentence, phrase, and document [9].

In [10], the authors utilized BERT to enhance Aspect-Based Sentiment Analysis (ABSA) accuracy in Persian. Their study aimed to improve ABSA performance in the Pars-ABSA dataset and leveraged a pre-trained BERT model. The Pars-BERT pre-trained model, combined with an NLIM auxiliary sentence, achieved the highest accuracy of 91% for an ABSA task on the Pars-ABSA dataset. This result represents a 5.5% absolute improvement over the previous best model.

Shangipour et al. proposed a new model for aspect-based emotion analysis (Pars-ABSA) in [11]. The Pars-ABSA dataset comprises 10,000 social media user reviews, with 5,114 negative, 3,061 positive, and 1,827 neutral reviews. The authors employed a Bi-LSTM neural network for emotion classification.

In [12], a combined approach for ranking prediction in Persian is presented. The proposed system performed better than the Naive Bayes algorithm and a dictionary-based method on a large dataset containing 16,000 Iranian customers. The primary method improved in this study was the dictionary-based approach used to analyze surveys in Persian. The results suggest that the proposed method is effective in accurately detecting polarity.

Hosseini et al. developed a Persian sentiment analysis corpus named SentiPers, which includes 26,000 annotations in Persian [13]. One of the critical features of SentiPers is the incorporation of both formal (written) and informal (verbal) sentences. The data used to build SentiPers was sourced from the Digikala website. The output tags are divided into six categories, ranging from -3 to +3, where -3 indicates a negative sentiment about the product, +3 represents the highest satisfaction, and zero represents a neutral sentiment.

One model that integrates four deep learning models (CNN, LSTM, Bi-LSTM, GRU) was used to address both Aspect Category Detection (ACD) and Aspect Category Polarity (ACP) tasks simultaneously, as described in [14]. While other studies have proposed separate solutions for these tasks, this study developed a consolidated solution for both tasks using the same model.

PerSent Persian sentiment lexicon, consisting of 1000 idiomatic expressions, was developed by Dashtipour et al. to aid word recognition in Persian texts and improve the accuracy of Persian classification. The reviews were labeled into positive and negative categories by three experts with an accuracy of 0.1 decimal places using a scale from -1 to +1. The CNN neural network was utilized to classify texts and evaluate the model on datasets such as Movie Reviews, Persian VOA, and Amazon reviews [15].

To collect Iranian users' reviews of movies and cinema news, caffecinema.com, and cinematicket.org were used and the reviews were tagged by three Persian-speaking individuals using the Persent dictionary [16]. The Stacked-bidirectional-LSTM neural networks were utilized for classification, and the bag of words method and PCA algorithm were used to reduce the data to 200 dimensions for feature extraction.

Sabri and her colleagues created a dataset of Persian-English code-mixed tweets and labeled the dataset. They introduced a model that utilizes BERT pre-trained embeddings, Yandex, and dictionary-based translation models to automatically learn the polarity scores of these tweets. This model was found to outperform baseline models that utilize Naïve Bayes and Random Forest methods [17].

III. PROPOSED METHOD

Fig. 1 displays the key stages of the suggested approach. The initial phase involves balancing the imbalanced dataset to avoid any bias towards a particular class in the model. In our experimental setup, the dataset used to train and test the models had imbalanced classes, with a significantly larger number of customers recommending a product than those who did not or

were uncertain. To ensure a fair representation of all three classes, we implemented an under-sampling technique that selected 10,535 reviews from each of the three classes for our research.

In the next step, the Hazm library [18] was used to normalize reviews. Normalization converts spaces to semi-spaces if necessary (such as turning می شود to می‌شود), and converts some letters to their correct Persian format (such as turning ي to ی).

Subsequently, stemming and lemmatization techniques were utilized to simplify a word to its basic form, which could be its root. For instance, stemming the word "interesting" could result in "interest". Due to its complex morphology, the Persian language involves concatenating prefixes and suffixes to words to change their meaning. Additionally, Persian nouns are suffixed to indicate possession and plurality, similar to English nouns. However, Persian verbs are subjected to more significant modifications than English verbs, varying based on tense, person, negation, and mood. Consequently, a given verb may exist in multiple forms and variations [19]. For this study, the Persian stemmer introduced by Taghva et al. [20] was employed.

In the next step, to remove stopwords, a file containing more than 2000 Persian stopwords was used [21]. These stopwords typically hold little significance in conveying the sentence's meaning, and their removal does not affect the sentence's comprehension. Examples of Persian Stopwords include "با", "از", "در", etc. This process streamlines the conversion of words to numerical values by reducing the overall number of words.

Next, we utilized two distinct techniques to vectorize the reviews. The first approach involved using the Bag-of-Words (BoW) method, which was utilized to train conventional machine learning algorithms such as Naive Bayes, Decision Tree, Random Forest, and *k*-nearest neighbor. Vectorization was conducted using the CountVectorizer method of the Scikit-learn Python library. The BoW model generated a matrix where the rows represented the reviews, and the columns contained all the words utilized in the reviews. For each review, the number of times each word appeared was computed, and the value was recorded in the corresponding column.

Another vectorization method applied in this research is the fastText word embedding [22] introduced by Facebook in 2016. fastText assigns a vector to each word to ensure that words with similar meanings are represented by the same vector.

To predict loyalty levels, two approaches were employed: conventional machine learning algorithms utilizing bag-of-words vectorization, and a deep learning model that uses fastText word embedding. The results of both methods were evaluated for comparison.

For predicting loyalty, a deep learning model incorporating convolution, max pooling, and LSTM layers was used. Fig. 2 illustrates the architecture of this model.

The deep learning algorithm was constructed using a sequential model, which creates a linear stack of layers, allowing for the addition of any number of layers to the network. This approach facilitates a layer-by-layer model. Initially, a one-dimensional convolution layer was established with 32 filters, which can be adjusted as needed, though it must always be a power of two.

In this research, the kernel size, or filter size, is set to 3, which is determined through a process of trial and error. The kernel is a convolution filter that applies specific modifications to the data as it moves across it. By reducing the dimensions of the input text, the kernel streamlines the training process for the model.

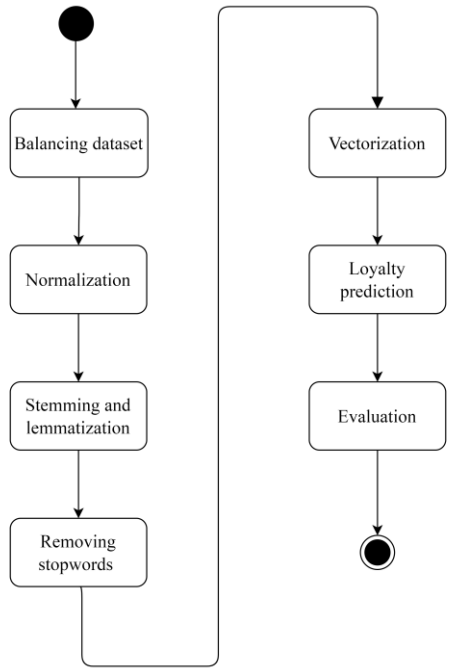

Fig. 1. Steps of the proposed method

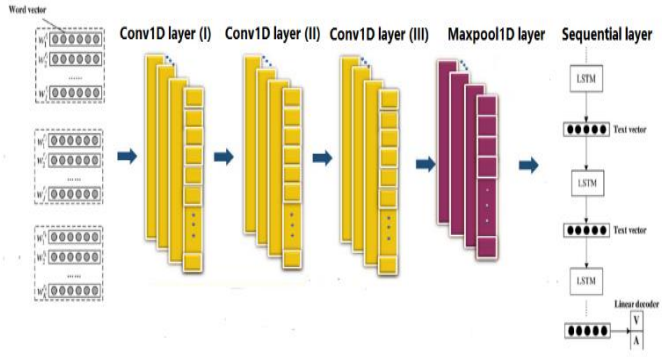

Fig. 2. Architecture of the deep learning-based model

The ReLU activation function is employed in our model. Activation functions act like gates in each neuron, receiving input from the previous layer's neurons (which are multiplied by their corresponding weights and then added to a constant bias value) and transferring the output to the next layer.

Popular non-linear activation functions used in our model include ReLU, Sigmoid, and Softmax. The activation function decides whether each neuron is activated or not. Only linear equations can be implemented without activation functions, which are not helpful when solving complex problems. Neural networks without activation functions are just linear regression models. In general, non-linear activation functions help models to create complex mappings between network inputs and outputs; in other words, these functions allow the model to adapt to complex and non-linear data. It is crucial for complex data, such as photos, text, video, audio, etc.

The number of layers of the deep neural network model is obtained by trial and error. In this model, there are three convolutional layers at the beginning. In the next step, a max-pooling layer is used. The primary purpose of using this layer is to make the output of the convolution layer smaller, and in our model, it converts every three elements into one element. The purpose of adding the max pooling layer is to reduce the computational load and memory.

The subsequent step involves implementing the LSTM neural network, which is configured with 512 neurons as its first parameter. Another significant parameter of the LSTM neural network is the dropout rate, which serves to prevent overfitting in the network. To address this potential issue, the dropout rate for this layer is set at 0.2.

Lastly, the dense layer is utilized. This layer is composed of a group of neurons, with each neuron receiving input from the previous layer's neurons. The initial parameter in this layer specifies the number of neurons, which is set at 512 for each dense layer. The final dense layer has three neurons because the model categorizes reviews into one of three classes with respect to the 3-class setting.

When working with a binary class setting, the number of neurons in the final dense layer must be two to correspond with the two possible classes. The activation function for this last dense layer is Softmax, whereas, for the previous dense layers, the activation function is Sigmoid. Once the layers and their parameters have been defined, the model is executed and evaluated.

IV. Experimental Results

A. Data Set Description

We applied an online shopping dataset from https://www.digikala.com/opendata/. The dataset consists of 100000 customer reviews and 12 features. The dataset features are “product_id”, “product_title”, “title_en”, “user_id”, “likes”, “dislikes”, “verification_status”, “recommend”, “title”, “comment”, “advantages”, and “disadvantages”. The “title_en” feature determines the product group, while the features, “likes” and “dislikes” are related to the number of likes and dislikes of each customer’s review. The “title” feature specifies the title of each customer’s review. The “advantage” and “disadvantage” features are the product's strengths and weaknesses submitted by the customer. As discussed earlier, only the “comment” and “recommend” columns were used in this research and 10535 reviews from each class are selected to train and test the models.

B. Evaluation Metrics

The metrics used for evaluating the loyalty prediction models are Precision, Recall, F1-score, and Accuracy, which are calculated based on the confusion matrix displayed in Table I using (1)-(4).

$$\text{Precision} = TP/(TP+FP) \quad (1)$$

$$\text{Recall} = TP/(TP+FN) \quad (2)$$

$$\text{F1-score} = 2*P*R/(P+R) \quad (3)$$

$$\text{Accuracy} = (TP+TN)/(TP+FP+TN+FN) \quad (4)$$

C. Evaluation Results

The Python programming language and Google Colab environment were employed to implement all algorithms in this study. The analysis of the results was conducted separately in two distinct settings. Firstly, the results of the 3-class setting, which includes the classes "recommending the product," "not recommending the product," and "having no idea," are presented. Next, the binary class setting results are displayed. For evaluation, 25% and 75% of the dataset were randomly selected for testing and training, respectively. The process of dividing the dataset into training and testing sets was repeated ten times, and the metrics' average values were reported. Tables II-VI display and compare the results of the 3-class setting.

In Fig. 3, the accuracies of the algorithms in the 3-class setting are shown.

The highest accuracy in Fig. 3 is achieved by the Naïve Bayes and Random Forest algorithms, while the *k*-nearest neighbor algorithm has the lowest accuracy. The results of our second approach with binary classes are presented and compared in Tables VII-XI.

The accuracies of the algorithms in the binary class setting are displayed in Fig. 4. The Naïve Bayes and Random Forest algorithms demonstrate better performance compared to the other tested methods. Furthermore, the results indicate a significant improvement in the algorithms' performances in the binary class setting.

TABLE I. CONFUSION MATRIX FOR CALCULATING EVALUATION METRICS

Actual Label		Predicted Label	
		Negative	*Positive*
	Positive	False Negative (FN)	True Positive (TP)
	Negative	True Negative (TN)	False Positive (FP)

TABLE II. NAÏVE BAYES RESULTS IN THE 3-CLASS SETTING

Naive Bayes	Precision	Recall	F1-Score
Recommend	0.70	0.73	0.71
No idea	0.55	0.52	0.53
Not Recommend	0.72	0.73	0.72
macro avg	0.66	0.66	0.66

TABLE III. DECISION TREE RESULTS IN THE 3-CLASS SETTING

Decision Tree	Precision	Recall	F1-Score
Recommend	0.62	0.62	0.62
No idea	0.45	0.45	0.45
Not Recommend	0.62	0.61	0.62
macro avg	0.56	0.56	0.56

TABLE IV. RANDOM FOREST RESULTS IN THE 3-CLASS SETTING

Random Forest	Precision	Recall	F1-Score
Recommend	0.69	0.74	0.72
No idea	0.56	0.49	0.52
Not Recommend	0.71	0.75	0.73
macro avg	0.65	0.66	0.66

TABLE V. K-NEAREST NEIGHBOR RESULTS IN THE 3-CLASS SETTING

KNN	Precision	Recall	F1-Score
Recommend	0.48	0.75	0.58
No Idea	0.45	0.32	0.37
Not Recommend	0.68	0.47	0.56
macro avg	0.53	0.51	0.50

TABLE VI. DEEP LEARNING RESULTS IN THE 3-CLASS SETTING

Deep Learning	Precision	Recall	F1-Score
Recommend	0.69	0.74	0.71
No Idea	0.48	0.42	0.45
Not Recommend	0.68	0.72	0.70
macro avg	0.61	0.62	0.62

TABLE VII. Naïve Bayes results in the binary class setting

Naive Bayes	Precision	Recall	F1-Score
Recommend	0.88	0.91	0.89
Not Recommend	0.91	0.88	0.89
macro avg	0.89	0.89	0.89

TABLE VIII. Decision tree results in the binary class setting

Decision Tree	Precision	Recall	F1-Score
Recommend	0.81	0.82	0.82
Not Recommend	0.82	0.80	0.81
macro avg	0.81	0.81	0.81

TABLE IX. RANDOM FOREST RESULTS IN THE BINARY CLASS SETTING

Random Forest	Precision	Recall	F1-Score
Recommend	0.89	0.89	0.89
Not Recommend	0.89	0.90	0.89
macro avg	0.89	0.89	0.89

TABLE X. K-NEAREST NEIGHBOR RESULTS IN THE BINARY CLASS SETTING

KNN	Precision	Recall	F1-Score
Recommend	0.71	0.89	0.79
Not Recommend	0.85	0.64	0.73
macro avg	0.78	0.76	0.76

TABLE XI. DEEP LEARNING RESULTS IN THE BINARY CLASS SETTING

Deep Learning	Precision	Recall	F1-Score
Recommend	0.87	0.85	0.86
Not Recommend	0.86	0.88	0.87
macro avg	0.86	0.86	0.86

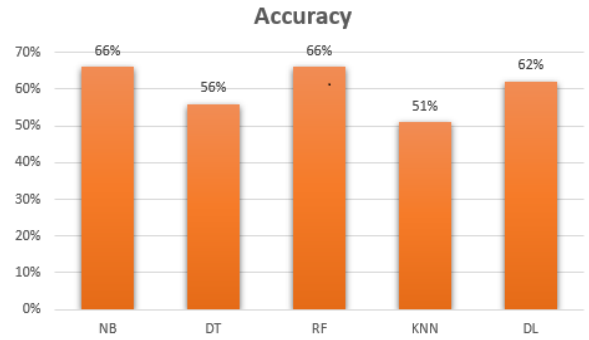

Fig. 3. Accuracies of the machine learning algorithms in 3-class setting

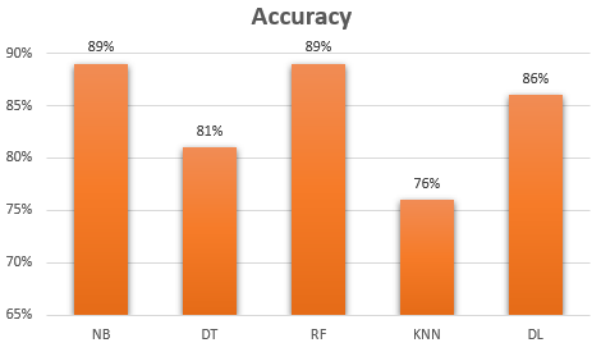

Fig. 4. Accuracies of the machine learning algorithms in the binary class setting

V. CONCLUSION AND FUTURE WORK

In this study, conventional machine learning algorithms and a deep learning model have been applied for customer loyalty prediction based on customer reviews in Persian. Regarding RQ1, the results imply that to predict customer loyalty, binary class setting is more effective than the 3-class setting. In both setting Random Forest outperformed other machine learning methods examined. The evaluation results indicate that the F1-score of the top-performing machine learning method is 0.89, therefore review analysis is effective to predict customer loyalty and the answer to RQ2 is positive.

The removal of stemming and lemmatization during review preprocessing led to a significant improvement in the accuracy of loyalty prediction models. This is likely due to the limitations of Persian stemming and lemmatization techniques.

Future work could explore the implementation of BERT language models for deep learning-based models, which may provide improved performance compared to fastText. Additionally, testing different deep learning architectures may also enhance the performance of the loyalty prediction model in the 3-class setting.

REFERENCES

[1] F. Buttle and S. Maklan, “Customer relationship management: concepts and technologies,” 2019.

[2] F. F. Reichheld, “The one number you need to grow,” *Harv. Bus. Rev.*, vol. 81, no. 12, pp. 46–55, 2003.

[3] A. Baquero, “Net Promoter Score (NPS) and Customer Satisfaction: Relationship and Efficient Management,” *Sustainability*, vol. 14, no. 4, p. 2011, 2022.

[4] S. Shumaly, M. Yazdinejad, and Y. Guo, “Persian sentiment analysis of an online store independent of pre-processing using convolutional neural network with fastText embeddings,” *PeerJ Comput. Sci.*, vol. 7, p. e422, 2021.

[5] H. Fazlollahtabar, “Intelligent marketing decision model based on customer behavior using integrated possibility theory and K-means algorithm,” *J. Intell Manag. Decis*, vol. 1, no. 2, pp. 88–96, 2022.

[6] P. Hanafizadeh, S. Mehri, and H. Hasanabadi, “Analyzing value creation in electronic retailing: The case of Digikala--Part A,” *J. Inf. Technol. Teach. Cases*, vol. 10, no. 2, pp. 72–82, 2020.

[7] A. Nazarizadeh, T. Banirostam, and M. Sayyadpour, “Sentiment Analysis of Persian Language: Review of Algorithms, Approaches and Datasets,” *arXiv Prepr. arXiv2212.06041*, 2022.

[8] K. Dashtipour, M. Gogate, J. Li, F. Jiang, B. Kong, and A. Hussain, “A hybrid Persian sentiment analysis framework: Integrating dependency grammar based rules and deep neural networks,” *Neurocomputing*, vol. 380, pp. 1–10, 2020.

[9] R. Dehkharghani and H. Emami, “A novel approach to sentiment analysis in Persian using discourse and external semantic information,” *arXiv Prepr. arXiv2007.09495*, 2020.

[10] H. Jafarian, A. H. Taghavi, A. Javaheri, and R. Rawassizadeh, “Exploiting BERT to improve aspect-based sentiment analysis performance on Persian language,” in *2021 7th International Conference on Web Research (ICWR)*, 2021, pp. 5–8.

[11] T. S. Ataei, K. Darvishi, S. Javdan, B. Minaei-Bidgoli, and S. Eetemadi, “Pars-absa: an aspect-based sentiment analysis dataset for Persian,” *arXiv Prepr. arXiv1908.01815*, 2019.

[12] M. E. Basiri and A. Kabiri, “HOMPer: A new hybrid system for opinion mining in the Persian language,” *J. Inf. Sci.*, vol. 46, no. 1, pp. 101–117, 2020.

[13] P. Hosseini, A. A. Ramaki, H. Maleki, M. Anvari, and S. A. Mirroshandel, “SentiPers: a sentiment analysis corpus for Persian,” *arXiv Prepr. arXiv1801.07737*, 2018.

[14] M. Vazan and J. Razmara, “Jointly modeling aspect and polarity for aspect-based sentiment analysis in persian reviews,” *arXiv Prepr. arXiv2109.07680*, 2021.

[15] K. Dashtipour, M. Gogate, A. Gelbukh, and A. Hussain, “Adopting transition point technique for persian sentiment analysis,” in *ICOTEN*, 2021.

[16] K. Dashtipour, M. Gogate, A. Adeel, H. Larijani, and A. Hussain, “Sentiment analysis of persian movie reviews using deep learning,” *Entropy*, vol. 23, no. 5, p. 596, 2021.

[17] N. Sabri, A. Edalat, and B. Bahrak, “Sentiment analysis of persian-english code-mixed texts,” in *2021 26th International Computer Conference, Computer Society of Iran (CSICC)*, 2021, pp. 1–4.

[18] M. Khallash and M. Imany, “Hazm: python library for digesting persian text: Github,” 2018.

[19] A. Bagheri, M. Saraee, and F. de Jong, “Sentiment classification in Persian: Introducing a mutual information-based method for feature selection,” in *2013 21st Iranian conference on electrical engineering (ICEE)*, 2013, pp. 1–6.

[20] K. Taghva, R. Beckley, and M. Sadeh, “A stemming algorithm for the

farsi language," in *International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II*, 2005, vol. 1, pp. 158–162.

[21] Kharazi, "Persian (Farsi) Stop Words List," 2021. [Online]. Available: https://github.com/kharazi/persian-stopwords.

[22] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguist.*, vol. 5, pp. 135–146, 2017.