WASSBERT: HIGH-PERFORMANCE BERT-BASED PERSIAN SENTIMENT ANALYZER AND COMPARISON TO OTHER STATE-OF-THE-ART APPROACHES

Masoumeh Mohammadi and Shadi Tavakoli

Department of Data Science & Machine Learning Telewebion, Tehran, Iran

ABSTRACT

Applications require the ability to perceive others' opinions as one of the most outstanding parts of knowledge. Finding the positive or negative feelings in sentences is called sentiment analysis (SA). Businesses use it to understand customer sentiment in comments on websites or social media. An optimized loss function and novel data augmentation methods are proposed for this study, based on Bidirectional Encoder Representations from Transformers (BERT). First, a crawled dataset from Persian movie comments on various sites has been prepared. Then, balancing and augmentation techniques are accomplished on the dataset. Next, some deep models and the proposed BERT are applied to the dataset. We focus on customizing the loss function, which achieves an overall accuracy of 94.06 for multi-label (positive, negative, neutral) sentences. And the comparative experiments are conducted on the dataset, where the results reveal the performance of the proposed model is significantly superior compared with other models.

KEYWORDS

Bidirectional encoder representations from Transformers (BERT), Bidirectional long short-term memory (Bi-LSTM), Comment classification, Convolutional neural network (CNN), Deep learning, Opinion mining(OM), Natural language processing (NLP), Persian language sentiment classification, Persian Sentiment analysis, Text mining.

1. INTRODUCTION

Watching movies is probably one of the most popular activities worldwide, and streaming movies online makes it more convenient. Furthermore, no one wants to waste their time on a film that is not worth watching [1]. Therefore, the Internet plays a crucial role in expressing opinions and sharing experiences about different movies. The goal of natural language processing (NLP) is to build a machine capable of understanding the contents of documents, including the contextual nuances of the language within them [2]. Sentiment analysis (SA) or opinion mining (OM) is a technique to determine the emotional tone. The SA models focus on polarity (positive, negative, neutral), feelings and emotions (angry, happy, sad, etc.), urgency, and even intentions (interested vs. not interested) [3].

Besides, it is widely applied to product reviews, social media, healthcare materials, etc. Many enhancements to SA models have been proposed in the last few years. In the next Section, we summarize and categorize some articles presented in this field that use various SA models such as machine learning (ML) algorithms or deep learning approaches. This paper aims to propose a

David C. Wyld et al. (Eds): ARIA, SIPR, SOFEA, CSEN, DSML, NLP, EDTECH, NCWC - 2022
pp. 209-220, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.121517

technique to classify reviews about movies depending on the sentiment they express, e.g., "The movie is surprising" (positive review), "I do not like cartoons" (neutral review), and "Crap, Crap and totally crap. Did I mention this film was total crap? Well, it's total crap" (negative review). Our main contributions to this study are as follows:

- The reliability of an SA solution depends highly upon obtaining sufficient data in Persian NLP. In this regard, the movie comments are crawled from several Persian websites. Then, data augmentation techniques are applied to the texts as described in Section 3 to generate additional and synthetic data.
- The next challenge is to deal with the imbalanced dataset complicated by the size, noise, and distribution. Most ML algorithms perform poorly and must be modified to prevent simply predicting the bulk of the data. Furthermore, metrics such as classification accuracy no longer make sense, and it is crucial to develop alternative techniques to evaluate predictions from imbalanced samples. Thus, several methods are performed to determine the best way to balance datasets; under-sampling appears most promising.
- Another foundational aspect of this study is the preprocessing phase, which among others, transforms comments, including emojis and emoticons, into plain text, using language-independent conversion techniques that are general and proper also to the Persian language.
- A customized list of stop words is devised to eliminate commonly used words. They carry very little helpful information, which improves the learning of the model keywords extracted as a reference for the global sentiment. Then, the attached label is transferred into Persian words as label embedding.
- We also conduct a comparative analysis of existing and proposed machine learning models and novel deep learning models regarding the recall, f1 score, precision, and accuracy.
- Our model adopts BERT-based word embedding to obtain each partial feeling and learn Persian sentences' complex and changeable structures. Finally, we use a custom loss function which results in our method outperforming the traditional and state-of-the-art models.

This paper organizes as follows. Section 2 includes the related works and a summary of the articles. A brief survey of comparable and benchmark methods is presented in Section 3, followed by the structure of the proposed approach. Consequently, the experimental setup and the results and evaluation are given in Section 4. Section 5 concludes and discusses future research.

2. RELATED WORKS

Many methods have been developed and tested around SA. They can be categorized as follows:

technique-based, text-oriented, level-based, rating level, etc. Collomb et al. [4] compared different points of view. From a technical viewpoint, they identified ML, lexicon-based, statistical, and rule-based approaches:

- The ML techniques perform learning algorithms to find the sentiment by training on a specific dataset.
- The lexicon-based method calculates sentiment polarity for a comment using the semantic sense or the semantic orientation of words and phrases in the review [4].

- The rule-based approach considers opinion words in content and then sorts them based on the number of positive and negative comments [4].
- Statistical models show reviews as a combination of hidden sights and ratings.

Another categorization based on the text structure includes document level, sentence level, or word/feature level classification. Reference [4] revealed that most techniques centralize a document-level classification. Also, the most current methods can identify sentiment strength for various aspects of a product/service and processes that intend to rate a review on a global level. Figure 1 depicts these details.

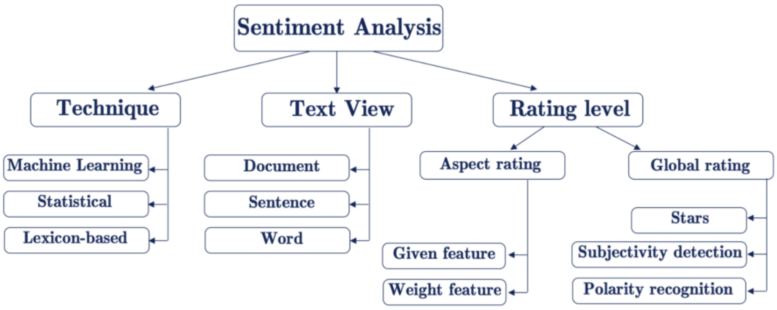


Figure 1. Types of sentiment classification, an overview of the classification techniques that have been used to answer the sentiment analysis questions.

Huifeng and Songbo [5] define several problems related to sentiment detection and discuss its different applications. They introduce semantic-based techniques, present ML methods, and mention two classification forms: binary (negative and positive) and multi-class (negative, neutral, and positive) sentiment classification.

Jakob & Gurevych [6] have focused on opinion extraction based on conditional random field (CRF). They apply a supervised methodology to a “movie review”. Toprak et al. [4] offer a scheme of annotation which contains two levels: sentence level and expression level. M. Hajmohammadi and R. Ibrahim [8] perform some ML techniques on a dataset of online Persian movie reviews to automatically classify them as either positive or negative. On this supervised classification task, they attain up to 82.9% accuracy.

Meanwhile, F. Amiri et al. [9] manually created a lexicon with sentiment scores and some rules on hand-coded grammar due to existing complexity, such as specific features, wrapped morphology, and the context-sensitivity of the script in the Persian language. They designed and developed a linguistic pipeline based on the framework and graphical development environment for robust NLP applications and named it GATE [10]. Their evaluation of the GATE pipeline reveals its overall accuracy of 69%.

Gonźalez et al. [11] design the BERT emotion detection tasks for TASS 2020 (an Albert-like model). It turns the highest accuracy in almost all the Spanish variants at three levels. Then Palomino and Ochoa [12] obtain the second-best result based on the BERT model. They apply an additional step of unsupervised data augmentation to improve their previous results for most variants of the Spanish language.

In [13], for Persian movie reviews, the deep learning model achieves 82.86% accuracy using the CNN model, obtaining significantly better results compared with previous models. Study [14] manually creates sentiment seeds to determine the polarity of a new lexicon. Their best accuracy is 81%. The proposed bidirectional LSTM network learning in [15] is considered the state-of-the-art model in Arabic SA. Their work improvement was 2.39% on average on the utilized datasets.

Amiri et al. [9] achieve an accuracy of 69% by SVM classifier for developing a lexicon to detect polarity on multi-domain products and movie reviews in Persian. Alimardani et al. [16] further improve this idea by proposing approaches that collected hotel reviews using an SVM classifier, achieving an accuracy up to 85.9%. Dos et al. [17] created a CharSCNN with two convolution layers to extract features and address SA. Wang et al. [18] developed a model based on LSTM to predict the sentiment polarities of tweets by composing word embeddings. Wu Xing et al. [19] demonstrated the subjective characteristics of the stock market by gated recurrent unit (GRU).

Nevertheless, the RNN can not be used in parallel calculations because of developing a gradient explosion. Vaswani et al. [20] offer a transformer to solve this problem and gain sustainable results in many NLP applications, including SA. Catelli et al. [21] use a multi-lingual technique based on BERT, performed a Named Entity distinction task for de-identification. Yu et al. [22] perform a BERT model to get state-of-the-art ancient Chinese sentence segmentation results.

3. Methodology

3.1. SVM

SVMs are the supervised learning methods for classification, regression, and outlier detection. This article uses an SVM from the Scikit-learn library as the first proposed model. It has been shown that the implementation of Gaussian kernels for SA is more performant than other nonlinear kernels.

3.2. BI-LSTM

We preprocess our dataset before feeding it to BI-LSTM. First, we normalize all comments using the Hazm normalizer [23]. The process of normalizing tokens returns them to their original form. Second, we separate each sentence into meaningful unit forms such as words, phrases, or subwords using the Keras tokenizer. Meanwhile, Hazm lemmatization is employed to merge two or more words into one by removing stop words from the penalties. The purpose of this step is to restore the roots of words or lemma, like می روم converted to رفت#رو. The Word2Vec training process vectorizes texts to help the system learn them. Fast-text is an NLP library developed by Facebook to use classification and word embedding [24]. Gensim Fast-text supports 157 languages. For the LSTM-based Persian SA, BI-LSTM is applied for the multi-label classification of movie reviews.

Since textual data are categorical variables, we need to convert them into numbers to feed the model [30]. One-hot encoding is an option to convert them into numbers. However, this approach is not viable due to its high memory demand. Meanwhile, the embedding layer is applied here to convert a word into a vector shape in multidimensional space and create a fixed-length vector to increase model efficiency. By using the max-pooling and dropout layer, we avoid overfitting problems. Global max-pooling reduces the dimension of the feature maps detected anywhere in this filter. For building the model, we compile the model with categorical cross-entropy loss function and Adam optimization. The model contained 5,535,003 trainable parameters. With 20 epochs, we run the BI-LSTM model and achieve the best mean accuracy of %87.01.

3.3. CNN

A CNN can extract multidimensional features (nonlinear features) without considering the probability of occurrence. There are 100 filters with a kernel size of 4, so each filter looks at a window of 4-word embeddings. It normalizes the previous layer's activation at each batch (batch normalization) by applying a transformation that maintains the mean activation close to zero and the activation standard deviation close to one [31]. After the activation function, a max-pooling layer is added.

3.4. BERT

Bidirectional encoder representations from transformers equip dense vector representations for NLP by using a deep, pre-trained neural network with the transformer architecture [16]. The original English language BERT has two models [16]:

1. the BERT-base: 12 encoders with 12 bidirectional self-attention heads.
2. the BERT-large: 24 encoders with 16 bidirectional self-attention heads.

There are also some other BERT models available:

- Small BERT: this model is a sample of the original BERT with a smaller number of layers [25].
- ALBERT: this is the "A Lite" version of BERT in which some of the parameters are reduced.
- BERT experts: setting off on a pre-trained BERT model and fine-tuning the downstream role produces efficient NLP tasks. It can increase the performance by starting from the BERT model that better aligns or transfers to the task at hand [32]. This collection is called "BERT expert" trained on different datasets and functions to perform better downstream tasks like SA, question answering, and all jobs requiring natural language inference skills.
- Electra: This is a pre-trained BERT-like model that plays a role as a discriminator in a setup resembling a generative adversarial network (GAN).
- ParsBERT: This model is pre-trained on large Persian corpora with more than 3.9M documents, 73M sentences, and 1.3B words [26].

3.4.1. The proposed BERT-based Model

We base our model on ParsBERT [26]. In this regard, 'HooshvareLab/BERT-FA-Base-uncased' was created, including 12 hidden layers and 12 attention heads. One dropout and a linear classifier with 768 hidden sizes. The whole model is displayed in Fig.2 Moreover, the parameters used in the proposed model are summarized in Table 1:

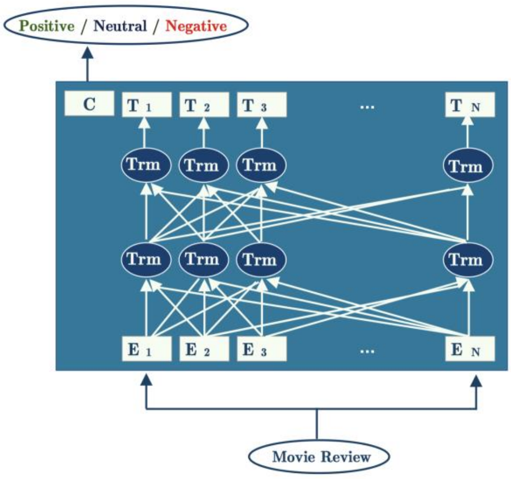


Figure 2. The BERT performs DL-based NLP tasks. It provides a model to understand the semantic meaning using NLP. The model uses movie comments as input and determines whether they are positive, neutral, or negative.

Table 1. The hyper-parameters that affect our purpose (feature importance) are empirically tested. Our experiments suggest that population-based training is the most efficient method for tuning the transformer model's hyper-parameters.

Parameter	Value
Epochs	3
Learning rate	2E-05
Train-batch-size	16
Valid-batch-size	16
Test-batch-size	16

3.4.2. Text classification using BERT

The following steps are followed in this investigation: Set up the Adam optimizer from transformers. Import and preprocess the dataset: The comments have different lengths. Detecting the most normal range could help us find the maximum length of the sequences for the preprocessing step. Create a BERT tokenizer: Tokenization separates a sentence into individual words. Besides, the inputs (users' movie reviews and comments) must be changed to numeric token ids and arranged in tensors before inputting to BERT [25]. It is a pre-trained model that has its input data format. Its structure contains two parts:

- The BERT summarizer that includes a BERT encoder and a summarizing classifier,
- The BERT classifier.

Figure 3 depicts both summarizing sectors.

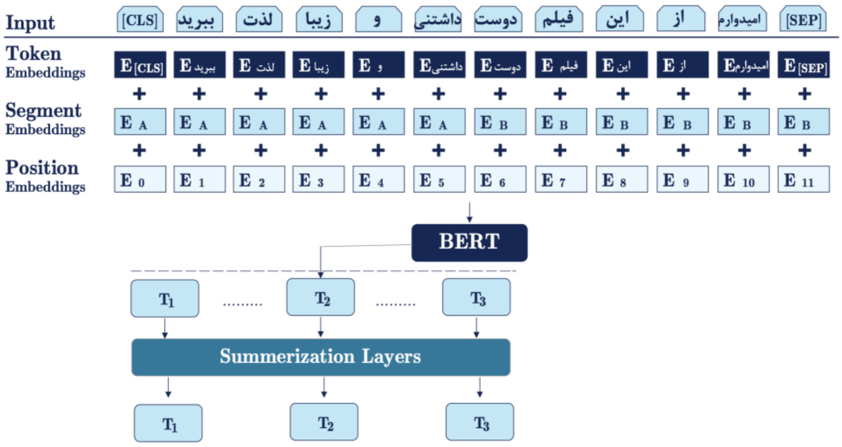
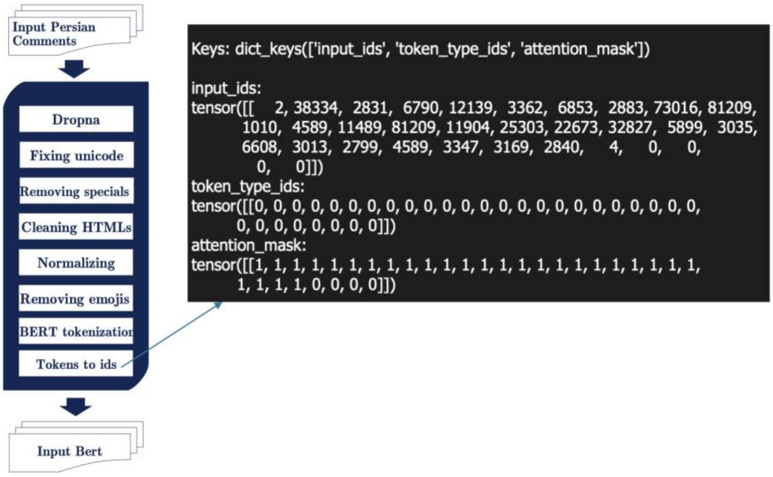


Figure 3. The BERT makes multiple embedding by a word to detect and report the content. Its input embedding includes the token, segment, and position components. The encoder gains the knowledge of interactions between tokens in the context, while the summarizing classifier learns the interactions between sentences.

The encoder learns the interactions among tokens in the document, while the summarization classifier learns the interactions among sentences. The BERT classifier has input and output. As figure 3 illustrates, [CLS] and [SEP] tokens separate two parts of the input. Each sentence is modeled as a sequence where the [CLS] token shows the beginning and [SEP] is a token to separate a sequence from a subsequent one [25]. After splitting words into tokens and converting the list of strings into a vocabulary index list, i.e., output for classification, we use the outcome of the first token, i.e., the [CLS] token. For more complicated results, we can use all the other token outputs. Figure 4 shows three outputs from the preprocessing that a BERT model would use. After this step, data is ready to convert to torch tensors and input to the BERT model. Figure 4 details the process: For NLP models to function, they need input in numerical vectors. Therefore, part of the process involves translating features such as vocabulary and parts of speech into numerical representations. Words can either be presented as uniquely indexed values (one-hot encoding) or as results from models such as Word2Vec or Fast-text, which match words with fixed-length feature embeddings. Each word has a fixed representation in these techniques regardless of the context; the words around them dynamically inform BERT representations of words. For example, consider the following two sentences: 1) “ آخرش که تیتراژ پایانی تمام میشه خیلی خنده داره”, which means: the ending is funny, and 2) “خنده ام میگیره” which means: it makes me laugh. Word2Vec produces the same word embedding for the word “خنده”(meaning laugh) in both sentences, while BERT’s word embedding for “خنده” would be different for each sentence. In addition to taking apparent differences such as polysemy, the context-informed word embeddings capture other forms of information that result in more accurate feature representations, making a better conclusion in model performance [27]. We use this advantage of BERT and some data augmentation techniques to increase the accuracy of this study. The dataset was divided into 22829 training, 2537 validation, and 2819 test sentences.



```
Keys: dict_keys(['input_ids', 'token_type_ids', 'attention_mask'])

input_ids:
tensor([[    2, 38334,  2831,  6790, 12139,  3362,  6853,  2883, 73016, 81209,
          1010,  4589, 11489, 81209, 11904, 25303, 22673, 32827,  5899,  3035,
          6608,  3013,  2799,  4589,  3347,  3169,  2840,     4,     0,     0,
             0,     0]])
token_type_ids:
tensor([[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0, 0, 0]])
attention_mask:
tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
         1, 1, 1, 1, 0, 0, 0, 0]])
```

Figure 4: Text inputs need to be transformed into numeric token ids and ordered in multiple tensors before being fed into the BERT; tokenization refers to assigning a sentence to single words.

There were three sentiment labels in the dataset (positive, negative, and neutral). The sample dataset is given in Table 2 below.

Table 2. The dataset contains 30000 user reviews which are balanced. A third of it owns negative comments labeled (-1), one third has the positive comments with the label (1), and the last part includes the neutral reviews tagged by (0).

Comments	Sentiment
بهترین فیلمی که تا حالا دیدم	1
شروع ضعیفی داشت امیدوارم در ادامه بهتر بشه	-1
تو رو خدا دوبله کنید	0

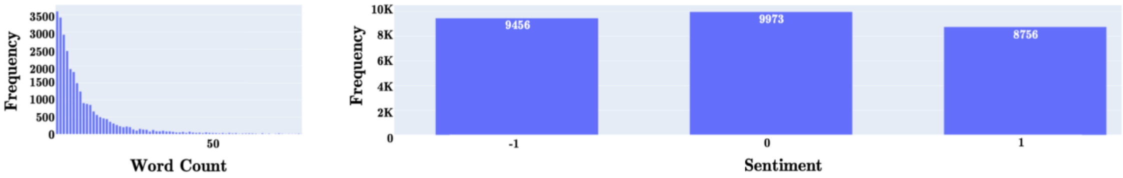


Figure 5. This chart illustrates the placement of the dataset before balancing. Balancing can be performed by over-sampling, under-sampling, class weight, or threshold. We use the under-sampling method to balance the dataset.

For the imbalanced dataset, two methods are applied: over-sampling and under-sampling. We observed better predictions in all deep models using the under-sampling technique [25]. Moreover, the clean dataset is augmented in two ways: random insertion and random swapping. The types of distribution of comments are demonstrated in Figs. 5.

The random swap does not work well in models due to existing particular characteristics in Persian, such as informal and conversational words, declension suffixes, various writing types, and word spacing. As a result, these traits affect Persian text accuracy. We also empirically observed that a delicately-crafted combination of Wasserstein and cross-entropy loss functions would result in significantly better model training. Consider the $X = \{x_0 , x_1 , ..., x_n \}$ to denote the possible outcomes or categories from the discriminator. Also, suppose $p : X \rightarrow [0, 1]$ and $q : X \rightarrow [0, 1]$ respectively denote the distributions for predicted and target values. The cross-entropy loss function (CLF) is then defined by:

$$H(p,q) = -\sum_{i=0}^{n} p(x_i)log(q(x_i)) \tag{1}$$

It is widely adopted as the loss function and a metric for the performance of classifiers. Recently, the Wasserstein metric has been showing excellent results, particularly in generative adversarial networks (GANs). This approach is often based on the Wasserstein-1 or Earth mover distance (EMD) between the two distributions, which basically measures the amount of mass needed to be transported to convert one distribution to another. Based on our notation, this distance is defined by:

$$W(p,q) = \inf_{\gamma\epsilon(p,q)} E_{(x,y)\sim\gamma}[||x-y||] \tag{2}$$

where Π(p, q) is the set of all join distributions having p and q as their marginal distributions. It can be shown using the Kantorovich-Rubinstein duality that this metric can be transformed to simply calculating the mean of a classifier's output [29]. Here, we propose to linearly combine the cross-entropy and Wasserstein loss function. The final loss function is of the form:

$$Finalloss = H(p,q) + \lambda W(p,q) \tag{3}$$

where the λ is the combination coefficient, which can be considered as a hyper-parameter. We empirically observed that the λ = n would be a good choice and set it to 3 for all our experiments. Table 3 below compares the results:

Table 3. The loss functions' comparison; combining cross-entropy and Wasserstein grants the best prediction compared with other Persian studies.

Loss function	Train-loss	Train-acc	Valid-loss	Valid-acc
Cross entropy	0.0373	0.989	0.284	0.927
Cross entropy+Wasseerstein	0.0317	0.991	0.219	0.940

4. EXPERIMENTAL RESULTS

Adjust the learning rate to about 2e -5 over three epochs. Using 16 GB of RAM and a Samsung SSD 870 500GB under the Ubuntu 64-bit operating system, the model was implemented in 11 minutes and 34 seconds under SA using an Intel Core i7 3.80GHz CPU with 16 GB of RAM. We developed machine learning and artificial intelligence projects with the visual intelligence model and Python libraries such as Numpy, Pandas, and Scikit-learn in Python 3.8.10. The data was collected from Persian movie review websites over 80 days, from 20 January 2020 to 25 April 2020.

4.1. Classifiers' measurement of this study

Different techniques are employed to extract the features from the movie reviews, and several opinions are applied to label the sentiments in the sentences. A balancing and augmentation method is used to carry the resulting dataset out, and each affected accuracy individually. As a result of completing various classifiers, performance metrics such as precision, recall, f1-measure, and accuracy [12] are calculated and reported in Fig. 6 and table 4. This figure reveals that BERT results are significantly higher than other algorithms.

Table 4. We compare several metrics to decide if a model performs well. The table shows the final result; the WassBERT gained the best scores.

Model	Accuracy	Recall	F1-score	Precision
SVM	75.5	75.5	75.5	80
BI-LSTM	87.01	82	85	82
CNN	83.38	80	81	83
WassBERT	**94**	**93**	**93**	**93**

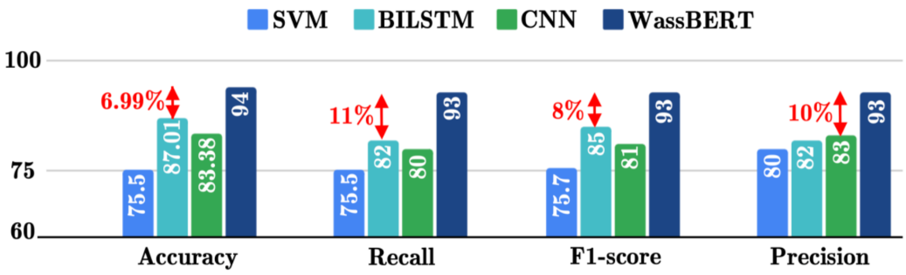


Figure 6. A comparison of WassBERT's performance metrics with those of other machine learning algorithms and deep models can be seen here where WassBERT is comparable to other Machine Learning algorithms and deep models in terms of its performance metrics. BERT yields an accuracy of 88.48 percent.

5. Conclusions

It is essential to recognize the sentiment of a movie comment in online reviews. However, the available Persian datasets are limited, and the existing models need to be improved. The proposed BERT model with a combination of Wasserstein and cross-entropy loss function is proved to achieve the best performance for the gathered Persian movie comments dataset. In a competitive study of deep learning models, proposed BERT's performance stands out (94%) among the deep learning models.

In future work, we address the dataset development in low resource languages, the balancing techniques, and augmentation methods that affect the model accuracy. We can also use explainable AI to Persian datasets with leading companies' data. Due to the lack of previous work on Persian datasets, our work cannot be compared to any previous ones and can now serve as a baseline for future work in this field.

REFERENCES

[1] Movie reason. [Online]. Available: https://www.everymoviehasalesson.com/blog/2021/9/4-reasons-to-read-movie-reviews

[2] The evolution of Natural Language Processing and its impact on the legal sector. [Online]. Available: https://www.lexology.com/library/detail.aspx?g=0facd988-1702-4850-92e2-2f4cd25ab9db

[3] Sharma, Ritu; Gulati, Sarita; Kaur, Amanpreet; and Chakravarty, Rupak, (2021) "Users' Sentiment Analysis toward National Digital Library of India: a Quantitative Approach for Understanding User perception". Library Philosophy and Practice (e-journal). 6372.

[4] A. Collomb, C. Costea, D. Joyeux, O. Hasan, and L. Brunie, (2014) "A study and comparison of sentiment analysis methods for reputation evaluation," Rapport de recherche RR-LIRIS-2014-002.

[5] H. Tang, S. Tan, and X. Cheng, (2009) "A survey on sentiment detection of reviews." Expert Systems with Applications, vol. 36, no. 7, pp. 10 760–10 773.

[6] N. Jakob and I. Gurevych, (2010) "Extracting opinion targets in a single-and cross-domain setting with conditional random fields. Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing," pp. 1035–1045.

[7] C. Toprak, N. Jakob, and I. Gurevych, (2010) "Sentence and expression level annotation of opinions in user-generated discourse. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics," pp. 575–584.

[8] M. S. Hajmohammadi and R. Ibrahim, (2013) "An SVM-based method for sentiment analysis in Persian language. International Conference on Graphic and Image Processing (ICGIP)," vol. 8768, p.876838.

[9] F. Amiri, S. Scerri, and M. Khodashahi, (2015) "Lexicon-based sentiment analysis for Persian text. Proceedings of the International Conference Recent Advances in Natural Language Processing,"pp. 9–16.

[10] H. Cunningham, (2002) "GATE: A framework and graphical development environment for robust NLP tools and applications. Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL)," pp. 168–175.

[11] J. Á. González-Barba, J. Arias-Moncho, L. F. Hurtado Oliver, and F. Pla Santamaría, (2020) "Elirf-upv at tass: Twilbert for sentiment analysis and emotion detection in spanish tweets. Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)," pp. 179–186.

[12] D. Palomino and J. O. Luna, (2020) "Palomino-Ochoa at TASS 2020: Transformer-based Data Aug mentation for Overcoming Few-Shot Learning. IberLEF@ SEPLN," pp. 171–178.

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

[14] N. Sabri, A. Edalat, and B. Bahrak, (2021) "Sentiment Analysis of Persian-English Code-mixed Texts. 2011 26th International Computer Conference, Computer Society of Iran (CSICC)," pp. 1–4.

[15] H. Elfaik et al., (2021)"Deep bidirectional lstm network learning-based sentiment analysis for Arabic text." Journal of Intelligent Systems, vol. 30, no. 1, pp. 395–412.

[16] S. Alimardani and A. Aghaie, (2015) "Opinion mining in Persian language using supervised algorithms. Journal of Information Systems and Telecommunication (JIST),".

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

[18] X. Wang, Y. Liu, C.-J. Sun, B. Wang, and X. Wang, (2015) "Predicting polarities of tweets by composing word embeddings with long short-term memory. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)," pp. 1343–1353.

[19] Wu, Xing and Chen, Haolei and Wang, Jianjia and Troiano, Luigi and Loia, Vincenzo and Fujita, Hamido, (2020) "Adaptive stock trading strategies with deep reinforcement learning methods., Information Sciences, vol. 538, pp. 142–158.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, (2017) "Attention is all you need." Advances in Neural Information Processing Systems. 31st Conference on Neural Information Processing Systems (NIPS), vol. 30.

[21] R. Catelli, F. Gargiulo, V. Casola, G. De Pietro, H. Fujita, and M. Esposito, (2020) “Cross-lingual named entity recognition for clinical de-identification applied to a covid-19 Italian data set,” Applied Soft Computing, vol. 97, p. 106779.

[22] J. Yu, Y. Wei, and Y. Zhang, (2019) “Automatic ancient Chinese texts segmentation based on BERT.” Journal of Chinese Information Processing, vol. 33, no. 11, pp. 57–63.

[23] Sobhe. Hazm. [Online]. Available: https://www.sobhe.ir/hazm

[24] Facebook. Fasttext. [Online]. Available: https://www.fasttext.cc

[25] J. D. M.-W. C. Kenton and L. K. Toutanova, (2019) “Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of NAACL-HLT,” pp. 4171–4186.

[26] M. F. M. M. Mehrdad Farahani, Mohammad Gharachorloo, (2019) “Parsbert: Transformer-based model for Persian language understanding,” Neural Processing Letters.

[27] K. K. Mnih A, (2013) “Learning word embeddings efficiently with noise-contrastive estimation. Proceedings of the Annual Conference on Advances in Neural Information Processing Systems (NIPS).” .

[28] S. I. C. K. C. G. D. J. Mikolov, T., (2013) “Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS), vol. 2, pp. 3111–3119. Curran Associates Inc., Lake Tahoe,”.

[29] M. Arjovsky, S. Chintala, and L. Bottou, (2017) “Wasserstein generative adversarial networks.” in International conference on machine learning. PMLR, pp. 214–223.

[30] Data Handling. [Online]. Available: https://towardsdatascience.com/data-handling-using-pandas-machine-learning-in-real-life-be76a697418c

[31] Gokhan Ciflikli, (2018) “Learning Conflict Duration: Insights from Predictive Modelling.” A thesis submitted to the International Relations Department of the London School of Economics for the degree of Doctor of Philosophy.

[32] Bert Expert. [Online]. Available: https://www.tensorflow.org/hub/tutorials/bert_experts

AUTHORS

Masoumch Mohammadi is the Co-Founder of Thumb Zone, a mobile usability testing platform company. A data scientist and application developer with over ten years of experience working with leading companies in social media. e-commerce, and online TV activities. She graduated with an M.S.C. in Artificial Intelligence and a B.S. in software engineering. Her interests include computer vision, natural language processing, and recommendation systems.

Shadi Tavakoli earned a Bachelor of Science in Electrical Engineering from Bu-Ali Sina University, Hamedan, Iran, in 2016. Currently, she is studying at the Islamic Azad University Central Tehran Branch for her Master's degree. Deep learning, natural language processing, and recommender systems are among her current research interests.