Iranian Journal of Science and Technology, Transactions of Electrical Engineering (2023) 47:1623–1634
https://doi.org/10.1007/s40998-023-00626-5

RESEARCH PAPER

Persian Text Sentiment Analysis Based on BERT and Neural Networks

Siroos Rahmani Zardak[1] · Amir Hossein Rasekh[1] · Mohammad Sadegh Bashkari[1]

Received: 13 January 2022 / Accepted: 23 March 2023 / Published online: 15 July 2023

Abstract

Sentiment analysis and opinion mining are studies that analyze people's opinions, feelings, experiences, and emotions through written language. This is one of the most active research fields in natural language processing and data mining, web mining, and text mining. In fact, because of its importance to business and society, it has expanded beyond computer science to management science and the social sciences. Sentiment analysis systems are used in almost every business and social area because opinions play an essential role in human activities and they are among our most influential behaviors. Our perceptions of reality and our choices are largely conditioned on how others see and evaluate the world. This is why when we have to make a decision, we often seek the opinions of others, and this is true not only for people but also for organizations. One of the challenges is the lack of a proper method for sentiment analysis in Persian, and algorithms that work in other languages such as English, French and Arabic but cannot be generalized to the Persian language. In this paper, Persian text sentiment has been analyzed using the BERT algorithm. Also, the efficacy of this algorithm has been evaluated by comparing the results with past works using different datasets. The experimental results are promising and indicate that the proposed approach significantly outperforms its counterparts in Persian text sentiment analysis accuracy. Our best results were obtained using ParsBERT and the original dataset, with an F1 score of 96.62 and accuracy of 94.38.

Keywords Sentiment analysis · Neural network · BERT

1 Introduction

SA[1] is an automated process of computationally understanding and classifying subjective information from source materials such as reviews on e-commerce websites, and posts/comments on social media platforms. One of the basic methods of SA is to assign positive or negative polarity to a text. SA is a subset of text mining and means finding the author's attitude towards a particular topic. In the era of the Internet, unstructured data is growing exponentially. People try to be seen and tend to express their opinions about products, news, and various issues in society.

Nowadays, the Internet makes it easy for users to share their views and discuss them. In addition, when a person wants to buy a product, he/she starts seeking other people's opinions and gathering information about it. These days, if one wants to buy a product, he is no longer limited to asking friends and family for comments. Online shops gather comments and discussions about every product from every perspective. As a result, a large number of people affect other people's future purchases, and businesses by commenting on each product (Sharami and Sarabestani 2020). For example, a software company can analyze comments about its recently updated app in a relative period to get feedback about that update. In recent years, we have seen that social media can have a direct impact on social and political views, for example, the revolution in Arabic countries in 2011 (Storck 2011). SA can be done in any field from consumer products to healthcare and

✉ Amir Hossein Rasekh
ahrasekh@zand.ac.ir

Siroos Rahmani Zardak
siroos.rahmani70@gmail.com

Mohammad Sadegh Bashkari
m.s.bashkari@gmail.com

[1] Zand Institute of Higher Education, Shiraz, Islamic Republic of Iran

[1] Sentiment Analysis

finance. Many works have been done in English, French and Arabic (Antoun et al. 2020; Martin et al. 2019; Pang et al. 2002) but there are limited works that have been done in Persian.

Farsi is an Indo-European language that is mainly spoken and written in Iran, Tajikistan, and Afghanistan. In Persian SA, the lack of strong tools for text processing, unavailability of a standard corpus, and also language structure is challenging (Saraee and Bagheri 2013). For example, there are multiple ways of writing some words; spaces are used, both in the words and between them and there are semi-spaces in some words. Using informal sentences and English characters in writing is another challenge. All of these challenges make it hard to tokenize the words properly. SA can be studied at three levels: document, sentence, and aspect (Liu 2012). In the document-level SA, we consider whether the entire text of the review is positive or negative (Pang et al. 2002). In sentence-level SA, the system usually determines whether a sentence feels positive, negative, or neutral. At the aspect level of SA, an entity and its characteristics are examined. For example, "The quality of the iPhone camera is awesome but its battery is not good"; the entity is iPhone and its characteristics are camera and battery, where sentiment with respect to camera is positive but towards the battery is negative.

In this paper, we focus on document-level SA, where we specify the polarity of a sentence by positive and negative.

The main purpose of this paper is document-level SA. Given that we want to examine the feeling of users' opinions about the products of online stores, this is done in the category of document-level SA. Figure 1 shows the steps in the proposed method which will be explained later.

The rest of the paper is organized as follows: In Sect. 2, the related work is reviewed, and Sect. 3 is devoted to the introduction of the proposed method. The experimental results are presented in Sect. 4. Finally, the paper is concluded in Sect. 5.

2 Related Works

We can divide the related work into three fields. The first category is works related to the English language. There are many articles in this category. The second category is works that are generalizable to several languages or, are language independent. Finally, the third category is Persian language-related works.

2.1 English Language-Related Works

There are several notable publications in this category. Among them, Pang et al. (2002) specified positive and negative polarity using machine-learning algorithms like support vector machine (SVM), naïve Bayes, and maximum entropy on movie review data. Maas et al. (2011) used a hybrid method based on WordVectors, LDA, and LSA to recognize document polarity. Oghina et al. (2012) used tweets from Twitter and comments of YouTube movie trailers as data and compared them to the IMDb rating of those movies. The results were close to the IMDb rating. Oppong et al. (2019) used data from three companies in Ghana. They annotate the data using three human annotators. As a label, they calculated the kappa coefficient between them. Finally, they used naïve Bayes for SA. The results were close to human sentiment.

2.2 Related Works in Other Languages and Multi-languages

Grave et al. (2018) proposed a model based on FastText, Glove, and Word2Vec for 157 languages. The model has been trained over Wikipedia and the Common Crawl internet archive. Su et al. (2018) proposed a method using Autoencode Neural Generative for multi-language words. Martin et al. (2019) proposed a BERT model for French. The results show that the model trained with 4 GB of data

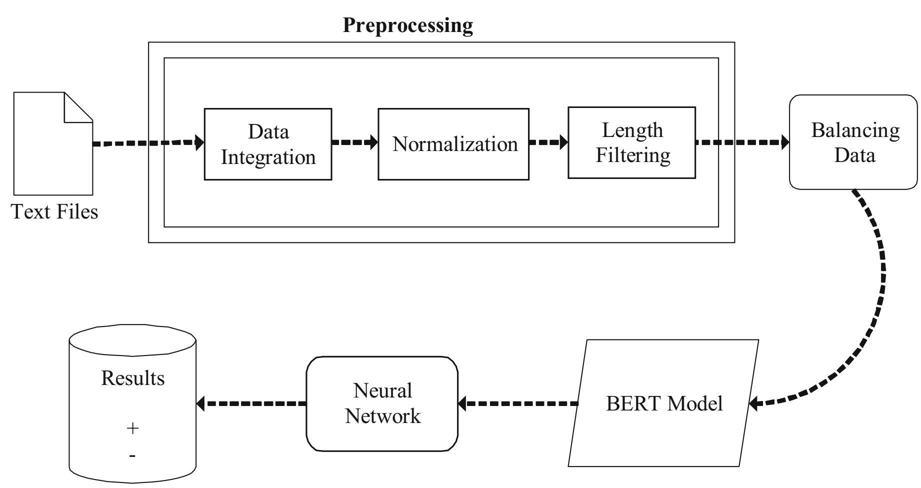

Fig. 1 Steps in the proposed method

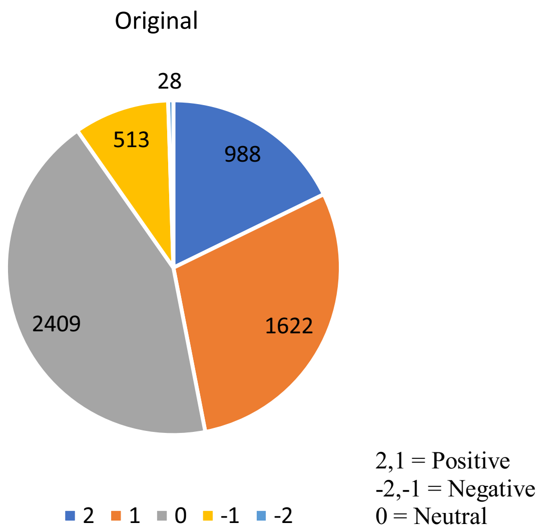

Fig. 2 Frequency of Original dataset labels

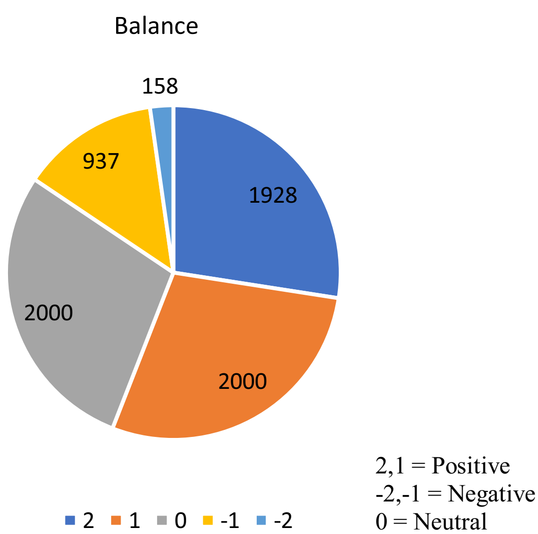

Fig. 3 Frequency of Balance dataset labels

outperforms the model trained with 138 GB of data. Antoun et al. (2020) proposed a BERT model for Arabic. The model is lighter than multilingual BERT by 300 Mb but the results show that the model performs better than multilingual BERT.

2.3 Related Works in Persian

Aleahmad et al. (2007) proposed a method using n-grams. The results show that Fourgram outperforms Trigram, Bigram, and Unigram. Ahmadi and Moradi (2015) proposed a hybrid model based on a rule-based approach and a Markov hidden model for the named entity recognition (NER) task. Basiri and Kabiri (2017) solved the lack of resources problem by proposing a lexicon that annotated

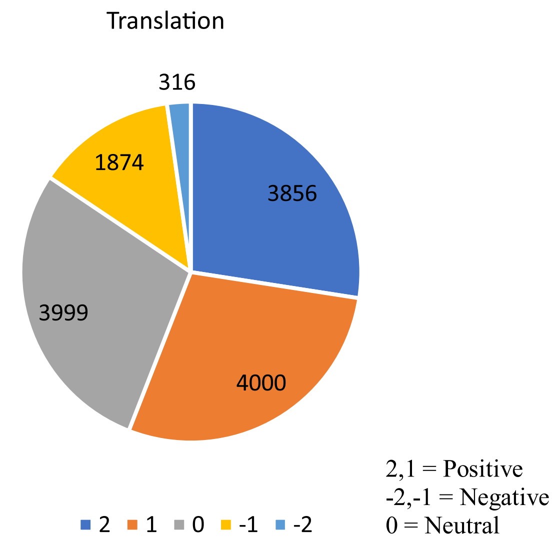

Fig. 4 Frequency of Translation dataset labels

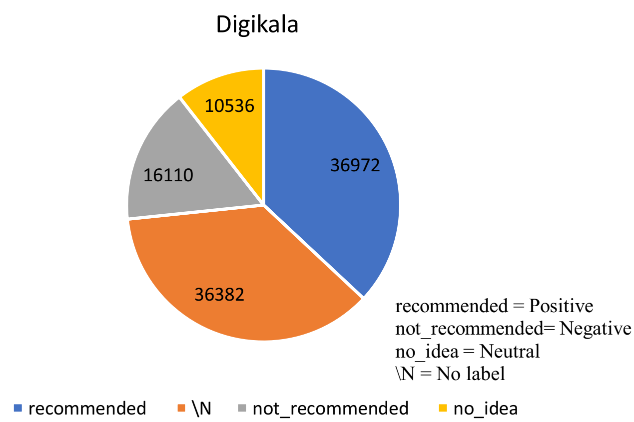

Fig. 5 Frequency of Digikala dataset labels

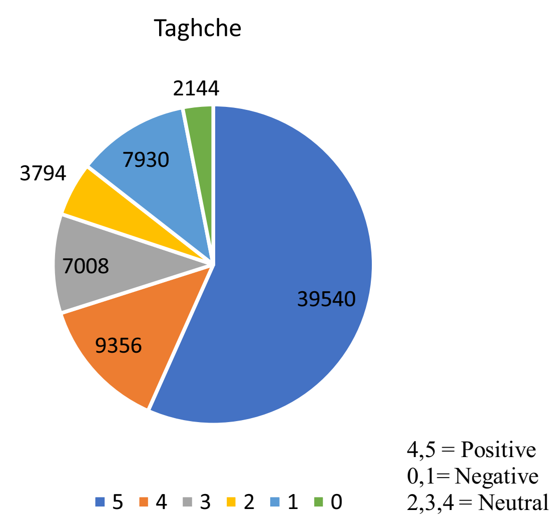

Fig. 6 Frequency of Digikala dataset labels

Table 1 Statistics of datasets

Dataset	No. of samples	Link to dataset
DeepSentiPers		https://github.com/JoyeBright/DeepSentiPers
Original	5560	
Balanced	7022	
Translation	14,045	
Digikala	100,000	https://www.digikala.com
Taghche	69,829	https://www.kaggle.com/datasets/saeedtqp/taaghche

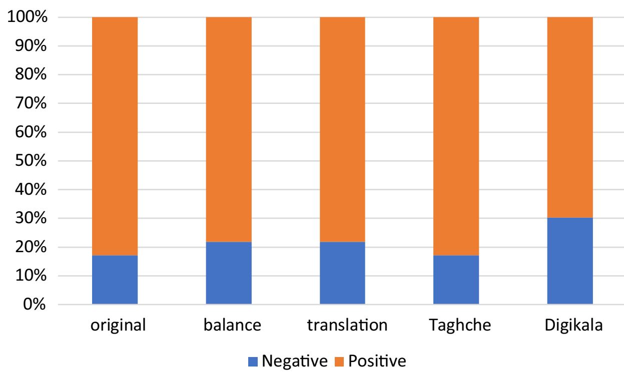

Fig. 7 Distribution of positive and negative labels in datasets after binarization

unlabeled data and then used naïve Bayes to obtain the polarity of short sentences. Ebrahimi et al. (2017) proposed a model that extracts adjectives in a sentence; then the polarity of that adjective is calculated using SentiWordNet. Razavi and Asadpour (2017) used a lexicon to detect the polarity of nouns and adjectives in the text. In another work, Dashtipour et al. (2018) proposed a feature extraction autoencoder model based on deep learning. Zahedi et al. (2018) compared embedding methods like Glove, Fast Text, and Word2Vec. The results showed that Fast Text works better in Persian text. Taher et al. (2020) used BERT for the NER task. Dashtipour et al. (2020) proposed a hybrid model using grammar-based rules and deep learning. Dastgheib et al. (2020) proposed a hybrid model based on structural correspondence learning (SCL) and deep learning. The SCL part selects features and sends them to the deep learning part of the model.

3 Proposed Method

Document-level SA is the main focus of this paper. The reason for doing this is that we want to examine how users feel about the products of online stores. This is done on the document-level SA.

3.1 BERT Model

BERT stands for bidirectional encoder representations from transformers. That means BERT can learn a language from both right to left and left to right. That helps us to understand a language better. As a result, BERT can be fine-tuned with an additional output layer (Devlin et al. 2018). The BERT model has two steps. The first step is pre-training on large unlabeled data. The second step is fine-tuning for a specific task.

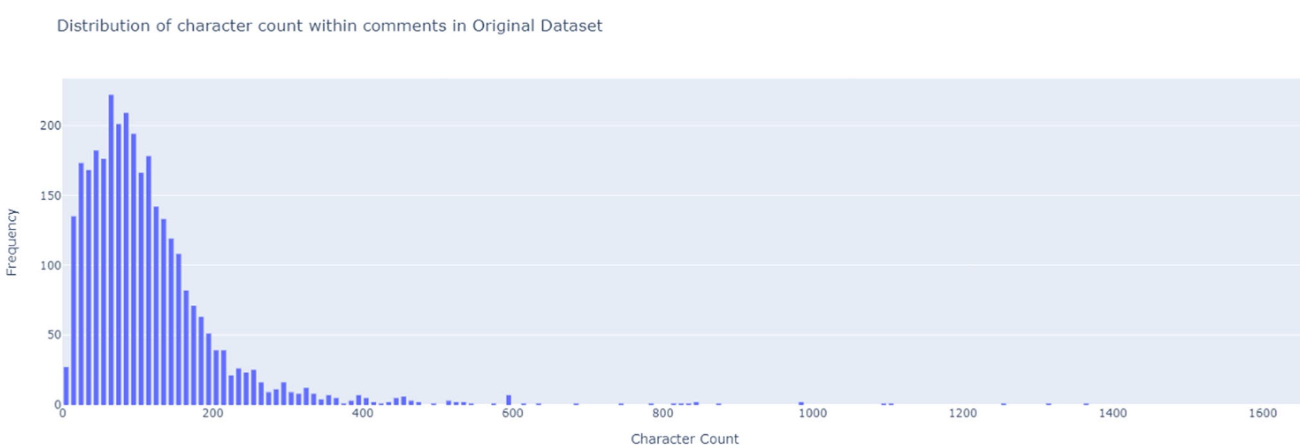

Fig. 8 Frequency of the length in Original dataset

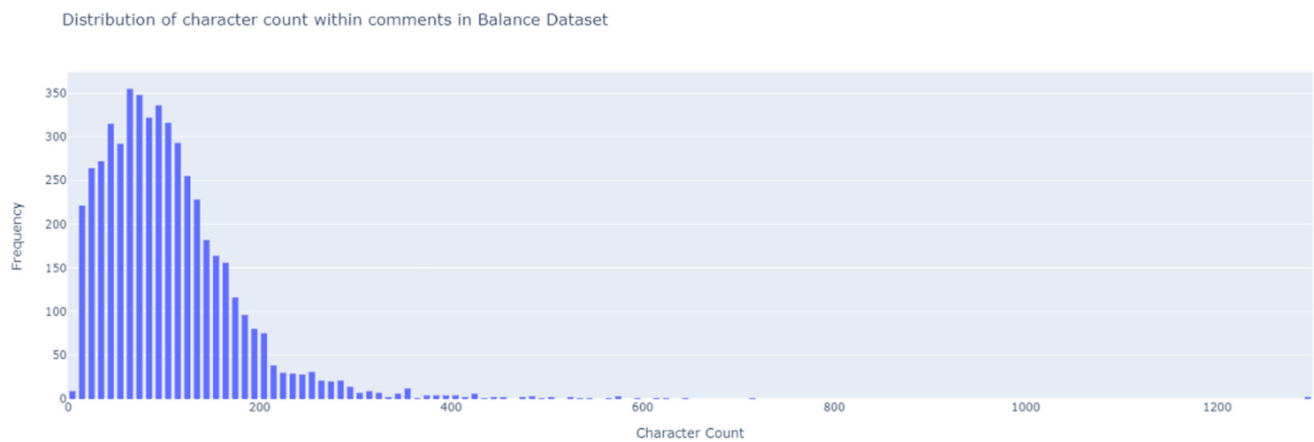

Fig. 9 Frequency of the length in Balance dataset

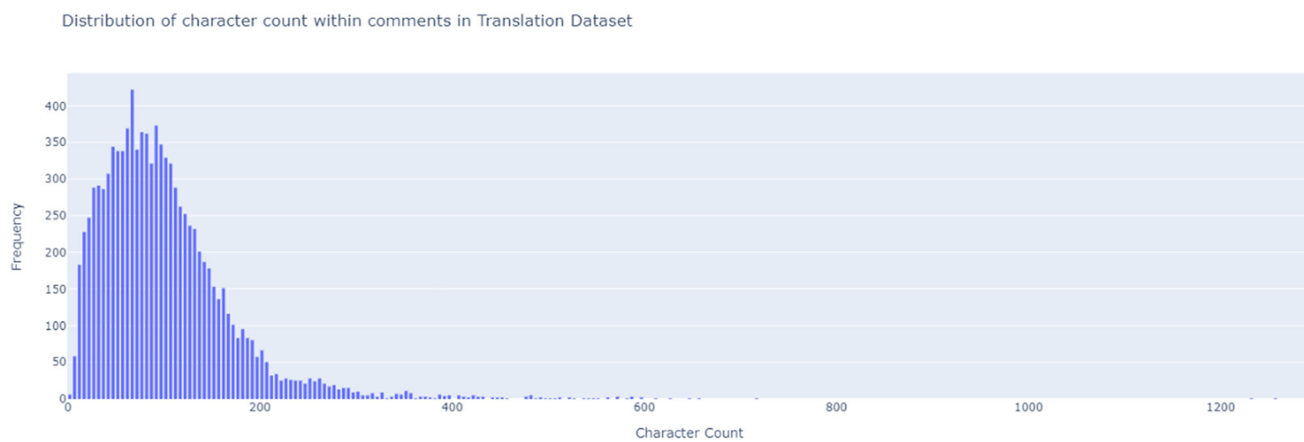

Fig. 10 Frequency of the length in Translation dataset

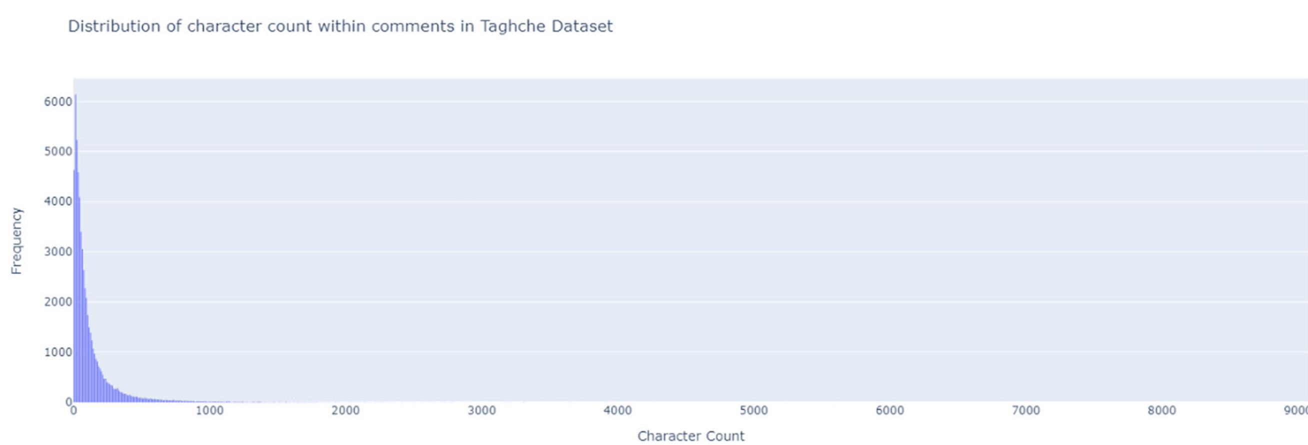

Fig. 11 Frequency of the length in Taghche dataset

3.2 Datasets

Choosing the right dataset for SA is one of the most important steps, and the final result depends on the quality of these data. In Sharami and Sarabestani (2020), three datasets are introduced that can be used to compare the performance of the proposed method. Furthermore, we have used two other datasets. The first dataset is Digikala[2]

[2] http://www.digikala.com.

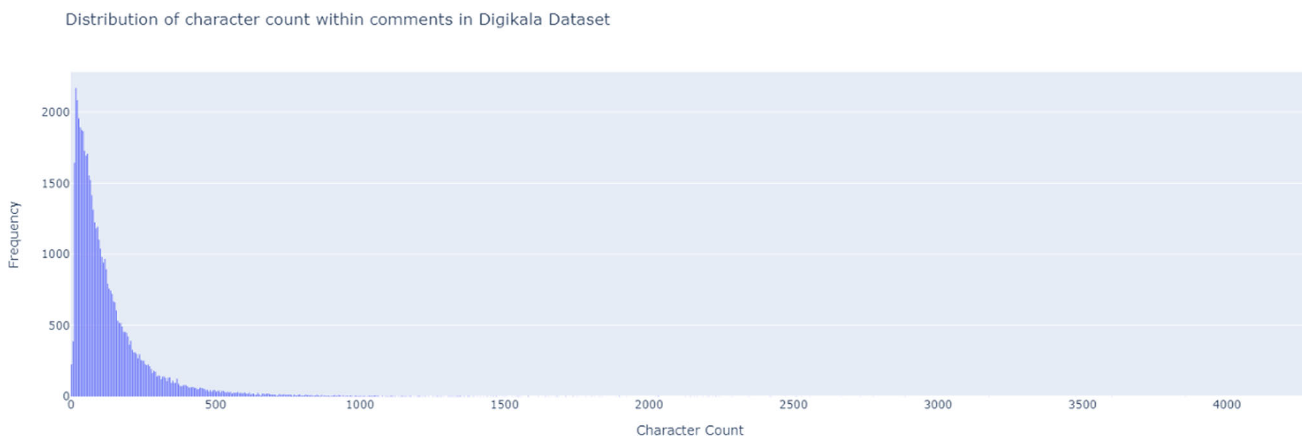

Fig. 12 Frequency of the length in Digikala dataset

open data[3] and the second one is Taghche dataset, which is available on the Kaggle website.

3.2.1 DeepSentiPers

DeepSentiPers consists of three datasets named original, balanced, and translation, and they have 5560, 7022, and 14,045 comments, respectively. The labels are the same, as they all have five integer labels ranging from −2 to 2. We ignore the neutral label zero. The negative labels are −2 and −1, and the positive labels are 1 and 2. Figures 2, 3, and 4 shows the original labels and frequency of these labels in DeepSentiPers datasets.

3.2.2 Digikala Open Data

This dataset consists of 100,000 user reviews on the Digikala website. The dataset has four labels, recommended as positive sentiment, not_recommended as negative sentiment, no_idea as neutral, and \N as no label. We ignored no_idea and \N labels and treated them as neutral and missing values respectively. Figure 5 shows the original labels and frequency of these labels in the Digikala dataset.

3.2.3 Taghche

This dataset consists of 69,829 book reviews from the taghche website[4]. This dataset has six integer labels ranging from 0 to 5. We assumed 0 and 1 as negative, 4 and 5 as positive, and ignored 2 and 3, and assumed them as a neutral labels. Figure 6 shows the original labels and frequency of these labels in the Taghche dataset.

[3] This publication was supported by the Digikala Open Data Mining Program.

[4] www.taaghche.com.

Statistics of datasets used in this research is presented in Table 1.

3.3 Train and Test Split

Although researchers in Sharami and Sarabestani (2020) used a 75/25 split, for validation, we used tenfold cross-validation in all of our experiments.

3.4 Preprocessing

As visualized in Fig. 7, the number of positive labels is much higher than negative labels. In most machine-learning algorithms, balanced classes lead to better results. We chose balanced samples from Digikala and Taghche datasets. Since the number of DeepSentiPers records is very low, and to enable a better comparison of their results, we did not balance these datasets.

Moreover, we observed that the length of many samples in the data was either too large or too small. As illustrated in Figs. 8, 9, 10, 11, and 12, the majority of samples have 12 to 120 characters. Therefore, we removed the samples that had fewer than 12 characters or more than 120 characters.

3.4.1 Dataset Label Integration

Since we used data from several sources, there were some irregularities in our data. For example, some records had integer labels and others had text labels. Therefore, we performed an integration phase on our data to address these irregularities.

3.4.2 Removing Punctuation

To prepare a text for machine-learning tasks, we removed some irregular characters from the text. These characters

Table 2 Characters with emotions

:P	😛	<3	🖤
;P	😜	:-(	☹
:,(	😢	:-)	😊
o.o	😮	:-|	😐
o_O	😳	;-)	😉
B-)	😎	-_-	😑
>_<	😣	:(	😞
3:)	😈	:D	😃

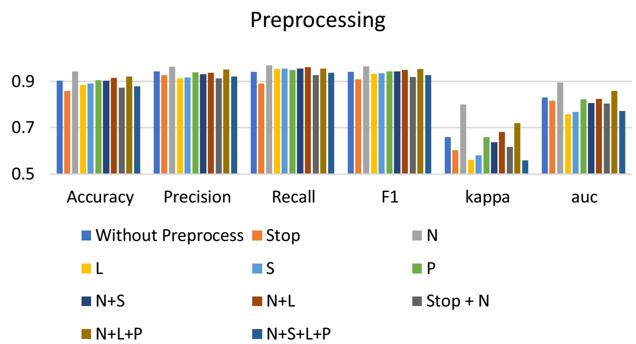

Fig. 13 Results of preprocessing algorithms (WP = Without Preprocessing, N= Normalizer, L = Lemmatizer, S= Stemmer, P = Punctuation Eraser, Stop = Stop Words Removal)

are like English characters, symbols, emojis, HTML, and JSON tags. Some of these characters are presented in Table 2.

3.4.3 Normalization

Semi-spaces are common in Persian words. A semi-space is a horizontal space with a length of zero between two Persian characters, which prevents two surrounding letters from sticking together. We used a tool called Hazm to deal with these spaces.

3.4.4 Lemmatization

Lemmatization is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's lemma, or dictionary form.

3.4.5 Stemming

Stemming is the process of reducing inflected words to their word stem, base, or root form, generally a written word form.

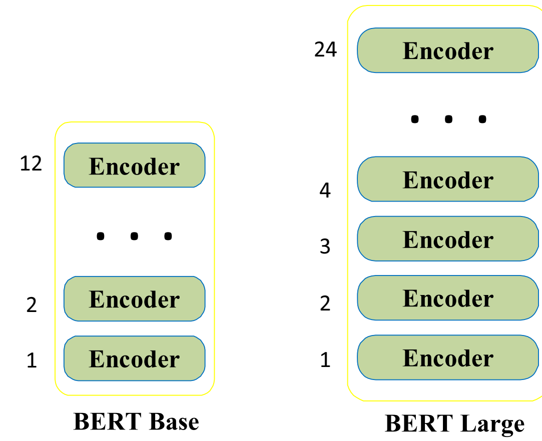

Fig. 14 BERT models

3.4.6 Stop Word Removal

Stop words are a set of commonly used words in a language. The purpose is to eliminate words that are so commonly used that they carry very little useful information.

According to Fig. 13, we observe that all of the preprocessing methods except normalization and punctuation removal have decreased the performance. When we combine normalization and punctuation removal, the performance is still lower than normalization alone. Therefore, we only used normalization.

3.5 BERT Models

BERT is implemented on the transformers architecture. There are currently two types of BERT models. The first model is the BERT base, which has 12 layers of encoders, 12 attention heads, and 110 million parameters. The second model is BERT large, which has 24 layers of encoders, 16 attention heads, and 340 million parameters. As can be seen in Fig. 14, this model uses only encoder blocks, and the decoder blocks that existed in transformers have been removed. We remove decoder blocks because we use a pre-trained model and this model is used for different tasks. After training the model, by adding a decoder, we can fine-tune the model for our work. In addition to SA, BERT can be used in areas such as next-word recognition, autoresponder systems, NER, and text classification. There is no need to train a new pre-trained model for each of these tasks. By pre-training one model, it can be used in many tasks.

We have used two pre-trained BERT models. The first model is proposed by BERT (Devlin et al. 2018) researchers and is called multilingual BERT. It used a machine translation Wikipedia dataset for training and it has been trained for 104 languages. This model has 12 encoder layers, 12 attention heads, 768 hidden states, and 110 million parameters. The second model is called ParsBERT (Farahani et al. 2020). It has been trained over Persian Wikipedia, BigBang website articles, Chetor website, Digikala mag, Eligasht, and Ted Talks subtitles. It has the same parameters as multilingual BERT.

3.6 Convert Text to Embedded Space

Since computers cannot understand text, we need a numerical means of text representation, which is called embedded spaces. The next step of our method is to convert text to the embedded space. One of the most important parameters in this section is MAX_SEQ_LENGTH, which is the maximum number of characters that the BERT model can store for an instance. If the number of characters in a sample is less than this parameter, it fills the remaining empty slots with zero, and if it is more, it takes only the value of this parameter from the first characters of that sample. The time it takes to train neural networks is directly related to the value of this parameter. If a large number is chosen, it might take days to train and requires a large amount of memory. To reduce time and memory consumption, we chose 120 for this parameter. In the first step, BERT converts text to tokens through a tokenizer. In the next step, it adds special tokens (CLS, SEP) to the start and end of tokens and then changes tokens with the corresponding ID of each token. The most important token is the CLS token, which has the embedding of the whole sentence. Figure 15 shows these steps.

After that, the computer then adds two more lists: one is attention_mask and the other is token_type_ids. Attention_mask tells us which words the model should pay attention to and which words it should not. If we pair two sentences together, token_type_ids show us which word matches which sentence. This part is not used in SA and is mostly used in areas such as question and answering systems. All these steps are performed by BERT's special tokenizer function.

In the final step, we train a neural network model using the embedded space of BERT output. After training the neural networks, our classification model is prepared. The type of neural network we used in this paper is deep learning. This is done using Keras in the TensorFlow library and the Python language using Google Colab.

3.7 Optimization

After preparing the pre-trained model, another challenge is the optimization of neural network parameters. One of the parameters is EPOCH. Each EPOCH occurs when all of our training data have moved forward or backward once in the neural network.

The optimized value for this parameter varies depending on what we do and the type of the data we use. We used a graph that is created during the training phase of the neural network. According to Fig. 16, the graph for 20 EPOCHs shows after the third EPOCH that there is no increase in accuracy, and also, in the loss section, there is a significant loss increase. As a result, we considered the EPOCH as 3.

For other optimization parameters, we used the Adam (Kingma and Ba 2014) algorithm, as the researchers in the BERT paper (Devlin et al. 2018) used this algorithm. Experimenting with different optimization techniques is beyond the scope of this paper, and we leave that to future works. Of course, according to Fig. 16, after the third epoch, overfitting occurs, and as a result the loss in validation is very high.

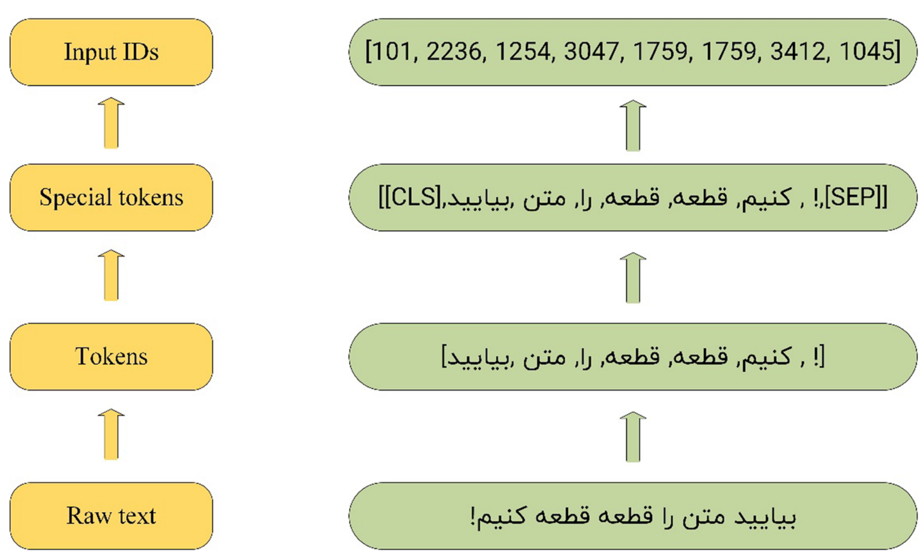

Fig. 15 Convert text to embedded space

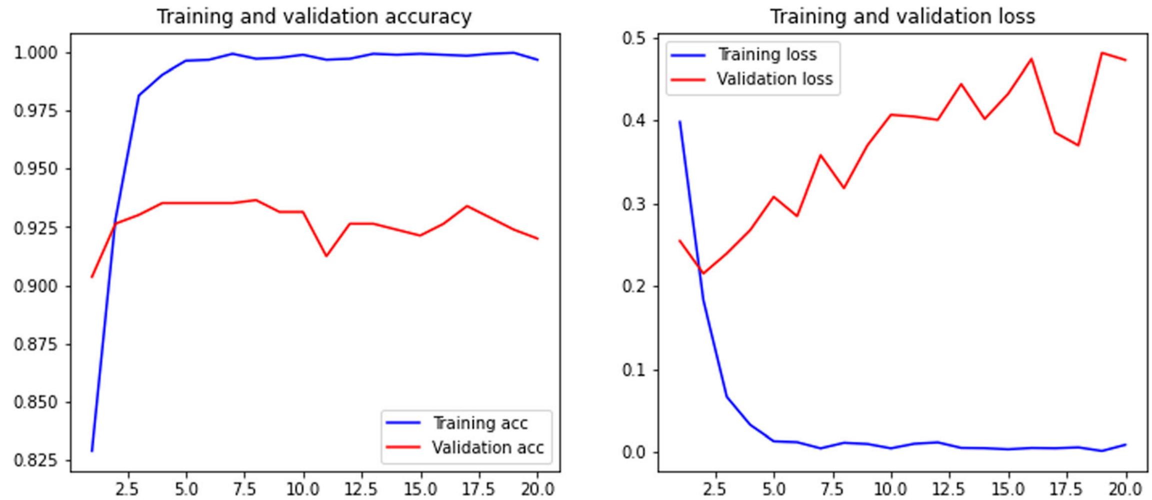

Fig. 16 Accuracy and loss graph for 20 EPOCHs

4 Evaluation

Although Sharami and Sarabestani (2020) only presented an F-measure score, we have used many evaluation metrics including accuracy, precision, recall, F-measure, kappa, and area under the receiver operating characteristic curve (ROC-AUC). Equations 1–6 used to calculate these metrics are as follows.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$F - \text{Measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{Kappa} = \frac{2 * (\text{TP} * \text{TN} - \text{FN} * \text{FP})}{(\text{TP} + \text{FP}) * (\text{FP} + \text{TN}) + (\text{TP} + \text{FN}) * (\text{FN} + \text{TN})} \tag{5}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{6}$$

Table 3 Comparing two BERT models and DeepSentiPers based on F-measure

Dataset name	BERT multilingual	ParsBERT	DeepSentiPers
Original	93.43	**96.62**	85.59
Balance	91.42	**95.34**	91.65
Translation	94.11	**96.02**	91.98

Bold values indicate the best results in comparison to other models

4.1 Results

In this section, we will review the results obtained from the proposed models. In Table 3, one can see that except for

Table 4 Comparing two BERT models

Dataset	Accuracy	Precision	Recall	F1	Kappa	AUC	Pre-trained model
Original	89.08	92.83	94.06	93.43	60.59	79.64	Multilingual
	94.38	**96.30**	**96.98**	**96.62**	**79.97**	**89.56**	ParsBERT
Balance	86.9803	94.29	88.85	91.42	64.50	84.81	Multilingual
	92.75	**95.62**	**95.11**	**95.34**	**78.96**	**89.77**	ParsBERT
Translation	90.80	94.14	94.11	94.11	72.94	86.43	Multilingual
	93.77	**95.91**	**96.14**	**96.02**	**81.66**	**90.76**	ParsBERT
Taghche	83.35	87.51	92.63	89.95	40.70	68.56	Multilingual
	85.92	**88.00**	**95.74**	**91.65**	**46.84**	**70.31**	ParsBERT
Digikala	**93.89**	96.46	**94.75**	**95.59**	85.64	93.32	Multilingual
	93.79	**97.91**	93.11	95.43	**85.74**	**94.21**	ParsBERT

Bold values indicate the best results in comparison to other models

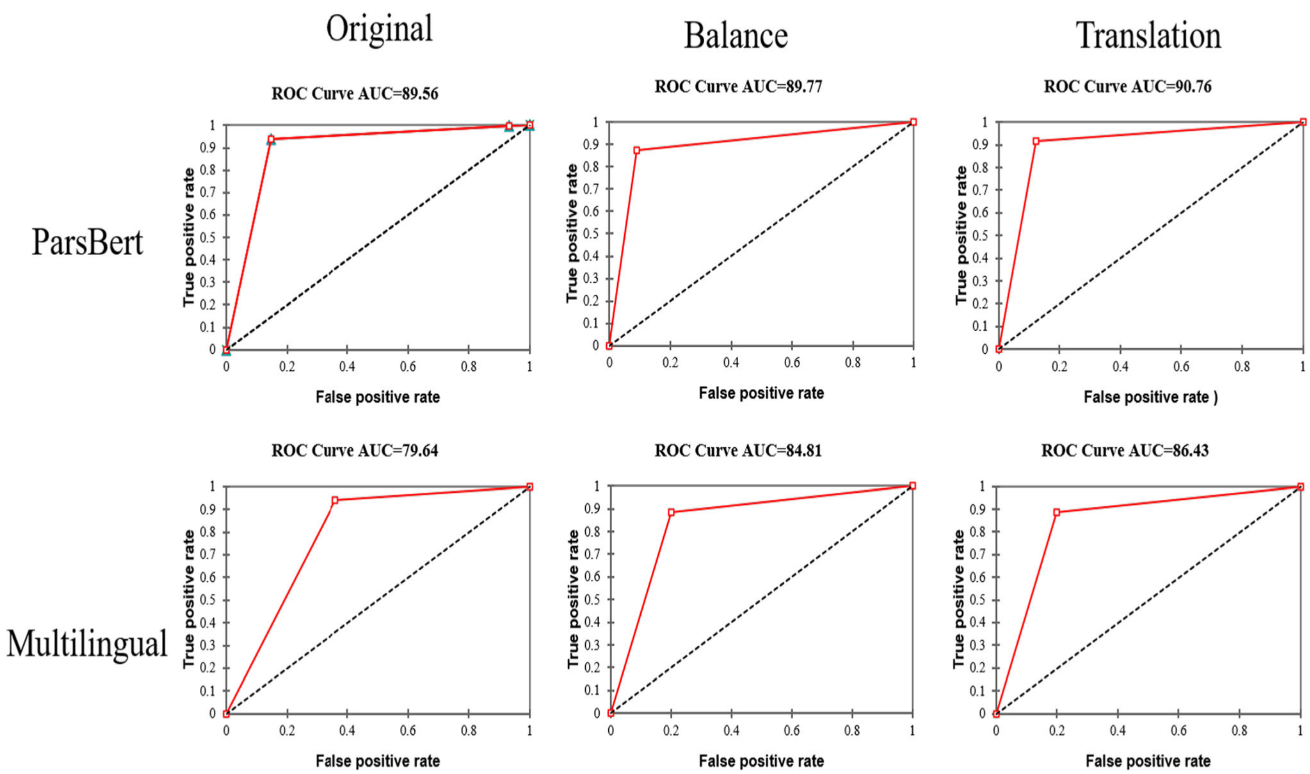

Fig. 17 ROC curve for both BERT models

the result of the multilingual model in the Balance dataset, both BERT models have achieved much better results.

As shown in Table 4, the ParsBERT model obtained better results. One reason for the better performance of this model is the pre-training of the model on Persian text, while the multilingual model translated English texts into Persian using machine translation and was then pre-trained.

AUC and kappa are the metrics that tell us how well we have separated positive and negative labels. As we discussed earlier, three of our datasets are imbalanced. If we look at the results, there is a significant gap between our BERT models in these metrics where datasets are imbalanced. According to Fig. 17, by looking at the ROC curve of these datasets, we can conclude that the ParsBERT model works significantly better in imbalanced data, and Table 5 shows that the *P*-value was very small in both models.

Table 5 *P*-value for imbalance datasets

Dataset	*P*-value	Pre-trained model
Original	< 0.0001	Multilingual
	< 0.0001	ParsBERT
Balance	< 0.0001	Multilingual
	< 0.0001	ParsBERT
Translation	< 0.0001	Multilingual
	< 0.0001	ParsBERT

Table 6 Error in *Taghche* dataset

Actual Label	Predicted Label	Text
1	0	لطفا کسایی ک خوندن نظر بدن ممنون
1	0	سوالاش کجاست پس ؟
0	1	من دارم کافکا در کرانه رو الان میخونم شناخت زیادی ندارم ازش اما کافکا در کرانه اش عالیه
0	1	نخوندم اما بنظرم خوبه . 😄

Table 7 Error in *Taghche* dataset

Actual Label	Predicted Label	Text
1	0	به نسبت کتابهای دیگر این نویسنده ، جذابتر است . ولی به اندازه ای که از اون تعریف شده بود منو جذب نکرد .

Table 8 Error in *Taghche* dataset

Actual Label	Predicted Label	Text
0	1	ترجمه عالی ، داستان عالی
0	1	این کتاب محشره فوق العاده است حتمابخونید ارزش خریدن وداره😍😍😍😍😍
0	1	هرکی تاحالا کتاب نخونده عاشق این کتاب میشه
0	1	کتاب بسیار عالی و ترجمه ی بسیار مناسب نسبت چند ترجمه ی موجود است .
1	0	ترجمه خیلی روان نیست
0	1	بسیار عالی
0	1	از خواندن نمونه کتاب فهمیدم کتاب زیبایست تشکر

4.2 Discussion

According to the results, the Taghche dataset has the worst performance. In this section, we will discuss some of the misclassifications in this dataset. According to Table 6, some of the samples had neutral sentiments, but the classifier has to classify them as positive and negative.

In Table 7, we observe that there are two sentences with different sentiments. The first one has positive and the second one has negative sentiments.

In our investigations, there were too many instances in the Taghche dataset that had false actual labels but the model had predicted them correctly. We present some of these sentences in Table 8.

5 Conclusion

In this paper, Persian text sentiment has been analyzed using the BERT algorithm. Additionally, the efficacy of this algorithm has been evaluated by comparing the results with past works using five datasets. The experimental results are promising and indicate that this proposed approach significantly outperforms previous methods. Based on the obtained results, we can conclude that BERT can also be used in Persian SA. The disadvantages of this model are high specification hardware requirements, the need for a pre-trained model, excessive training time, and the need for large amounts of unlabeled data. On the other hand, one important advantage is that, once trained on unlabeled data, the model can be used in various tasks such as next word prediction, NER, and SA. The use of standard and good-quality datasets has a significant effect on the final results and can be mentioned as the most important factor.

6 Future Works

With the availability of high-performance hardware resources, a new model can be pre-trained on various user review databases to be used in user-review SA. Using various optimization techniques, we can find the best value for the parameters in BERT as well as neural networks and achieve better results. Manual corrections of Taghche dataset labels can also provide a good-quality dataset for use in future works.

References

Antoun W, Baly F, Hajj H (2020) Arabert: transformer-based model for arabic language understanding. arXiv preprint arXiv:2003.00104..

Ahmadi F, Moradi H (2015) A hybrid method for Persian named entity recognition. In: 2015 7th conference on information and knowledge technology (IKT). 2015. IEEE.
Aleahmad A, Hakimian P, Mahdikhani F, Oroumchian F (2007) N-gram and local context analysis for Persian text retrieval. In 2007 9th international symposium on signal processing and its applications, pp 1–4. IEEE
Basiri ME, Kabiri A (2017) Sentence-level sentiment analysis in Persian. In 2017 3rd international conference on pattern recognition and image analysis (IPRIA). 2017. IEEE
Dashtipour K, et al. (2018) Exploiting deep learning for Persian sentiment analysis. In: International conference on brain inspired cognitive *systems*. 2018. Springer
Dashtipour K, Gogate M, Li J, Jiang F, Kong B, Hussain A (2020) A hybrid Persian sentiment analysis framework: Integrating dependency grammar based rules and deep neural networks. Neurocomputing 380:1–10
Dastgheib MB, Koleini S, Rasti F (2020) The application of deep learning in Persian documents sentiment analysis. Int J Inf Sci Manag (IJISM) 18(1):1–15
Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
Ebrahimi Rashed E, F., & Abdolvand, N. (2017) A supervised method for constructing sentiment lexicon in Persian language. J Comput Robot 10(1):11–19
Farahani M, Gharachorloo M, Farahani M, Manthouri M (2020) Parsbert: Transformer-based model for Persian language understanding. Neural Process Lett 53:3831–3847
Grave E, et al. (2018) Learning word vectors for 157 languages
Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Liu B (2012) Sentiment analysis and opinion mining. Synth Lect Hum Lang Technol 5(1):1–167
Maas A, et al. (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies
Martin L et al. (2019) Camembert: a tasty french language model.
ParsBERT Transformer-based Model for Persian Language.pdf.
Oghina A, et al. (2012) Predicting imdb movie ratings using social media. In: European conference on information retrieval. Springer, Berlin
Oppong SO, Asamoah D, Oppong EO, Lamptey D (2019) Business decision support system based on sentiment analysis. Int J Inf Eng Electron Bus 12(1):36
Pang B, Lee L, Vaithyanathan SJAPC (2002) Thumbs up? Sentiment classification using machine learning techniques
Razavi SA, Asadpour M (2017) Word embedding-based approach to aspect detection for aspect-based summarization of Persian customer reviews. In: Proceedings of the 1st international conference on internet of things and machine learning. 2017.
Saraee M, Bagheri A (2013) Feature selection methods in Persian sentiment analysis. Natl Lang Process Inf Syst 2013:303–308
Sharami JPR, Sarabestani PA, Mirroshandel SAJAPA (2020) Deepsentipers: Novel deep learning models trained over proposed augmented Persian sentiment corpus
Storck M (2011) The role of social media in political mobilisation: A case study of the January 2011 Egyptian uprising. University of St Andrews, Scotland
Su J, Wu S, Zhang B, Wu C, Qin Y, Xiong D (2018) A neural generative autoencoder for bilingual word embeddings. Inf Sci 424:287–300
Taher E, Hoseini SA, Shamsfard MJAPA (2020) Beheshti-NER: Persian named entity recognition Using BERT. 2020.
Zahedi MS, et al. (2018) Persian word embedding evaluation benchmarks. In: Iranian conference on electrical engineering (ICEE), 2018. IEEE