IJICTR
International Journal of Information & Communication Technology Research

Volume 15- Number 2 – 2023 (49 -58)

ChatParse: A New Application for Persian Bourse Chatbot

Ali Shahedi
Department of Electrical Engineering
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
alishahed@aut.ac.ir

Sanaz Seyedin*
Department of Electrical Engineering
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
sseyedin@aut.ac.ir

Received: 27 September 2022 – Revised: 17 October 2022 - Accepted: 3 January 2023

Abstract—**In this paper, we design and develop a brand new application for Persian stock-market chatbot using the retrieval approach namely ChatParse. The proposed architecture for this system consists of the Persian version of the BERT called ParsBERT in which we also add fully-connected and softmax layers to consider the number of classes according to our designed dataset. We manually design an appropriate Persian dataset for bourse application including 17 classes because we have found no Persian corpus for this application. ChatParse is able to have multi-turn conversations with users on the stock-market topic. The performance of the proposed system is evaluated in terms of accuracy, recall, precision, and F1-score on validation set. We also examine our application with test data acquired from users in real time. The average accuracy of the validation set over 17 classes is 68.29% showing the effectiveness of ChatParse as a new Persian Chatbot.**

Keywords: ChatParse, Persian Chatbot, stock-market, ParsBERT, attention.

Article type: Research Article

Publisher: ICT Research Institute

[DOI: 10.61186/itrc.15.2.49]

I. INTRODUCTION

Intelligent Conversational Agents (Chatbots) are mainly used as an assistant guide to help applications operate more desirably in terms of time and cost. It is cost-effective if Chatbots could fulfill human's duty, specifically in huge applications which need effective responses to user's utterances. Generally, there are two approaches for the open-domain chatbots to produce responses [1]:

1) Retrieval: This approach's algorithm works such that the desired sequence is given as the input to the model, and then the response corresponding to the top-rank class is selected. The model is not capable of generating a new response; instead, it uses responses from the database to produce an answer for the question or utterance of the user.

2) Generative: In this approach, the model generates responses based on Seq2seq (sequence to sequence) models; that is, with the help of specific

* Corresponding Author

algorithms, it generates a new answer that should be relevant to the user's utterance. The decoder path should apply a model such as the Recurrent Neural Network (RNN) [2] algorithm to generate a new response.

A problem that occurred because of the long sequence of the RNN model is the Vanishing/Exploding gradient. In this case, the model's gradient either becomes extremely small (converges to zero) or diverges to infinity. To address this issue, the authors of [3] proposed and compared the Gated Recurrent Unit (GRU) and Long-short-term memory (LSTM) model. These algorithms could control how the model processes information via the internal structure called gates. With these structures, the vanishing/exploding gradient problem is resolved. Input representation is an essential process in the training procedure. The sequence arrangement in terms of readability and ease is important because the sequence model only understands numerical sequences, which should be apprehensible for the model. The basic representation of sequence is One-Hot Encoding [4]. In this method, each token is considered the vector in which all indexes are zero except the one corresponding to the index value. In this method, the length of each vector is equal to the length of the vocabulary dictionary. One-hot encoding could not understand the relevance of the tokens because it represents the sequence in a biased way. Word Embedding [4] is the outstanding representation that connects the exact words. In this method, each word is defined in a lower-dimensional space toward one-hot encoding; i.e. each index in the vector shows the similarity of the corresponding feature to vectorized word.

One of the most influential models which improve the RNN model is the Attention model [5]. In this mechanism, the model attends specific information into the algorithm. Multi-head attention [5] is the method that consists of multiple attention computations. Each part of the computation attended in parallel is called the Head. Each attention model's output is concatenated, and the final value has resulted in a linear transformation. The Transformer [5] is a robust architecture that is based on the self-attention mechanism.

Finally, the worthwhile language model based on the Transformer, which improves the language models, is called BERT (Bidirectional Encoder Representations from Transformers) [6]. This model is finetuned on a massive dataset and usable in any NLP (Natural Language Processing) task. The BERT model is useful in many tasks such as sentiment analysis, text classification, and named entity recognition.

In this paper, we design and develop an application of the chatbot for stock market platforms (Bourse) in Persian which is able to perform multi-turn conversations. We use the language model transformer-based called the ParsBERT [7] as a Persian version of BERT in our application. This language model provides fewer challenges for designing our architecture. To improve the results, we also propose adding fully-connected and softmax layers to the ParsBERT. We find the best model in this application by changing the parameters and number of layers in terms of the loss and accuracy metrics. One of the most important contributions of this paper is designing an appropriate dataset for Persian Bourse Chatbot application because we could not find any specific Persian data for it. This dataset contains 590 questions and equivalent answers related to the stock market category in 17 classes. In this paper, we fine-tune the model to achieve the final results. Another contribution of this work is designing a brand new Graphical User Interface (GUI) application called the ChatParse to evaluate our model functionality.

This paper is organized as follows. In Section II, we talk about some related works. Section III presents our proposed approach. We describe our designed dataset in Section IV. Experiments and discussions are presented in Section V. Finally, we conclude the paper in Section VI.

II. Related Works

In this paper, we are inspired by several models and chatbot applications, and we have organized recent works based on these three approaches:

1) Many projects which are based on models have been practical in improving the performance of chatbots. In [2], the RNN algorithm was introduced, the first and preliminary model for designing seq2seq models. Then, due to only left-to-right training problem in RNN which cannot see the whole context, the Bidirectional RNN has been proposed in [8]. One step further, LSTM and GRU [3] have improved the RNN model. These two models are more precise than RNN because of the internal gates mechanism in their relevant algorithms. Word2Vec [4] algorithms such as CBOW [4] and Skip-gram [4] have rendered dense vector representations of tokens in sequence. The Transformer [5], which uses attention mechanisms in its architecture, has composed effective pre-trained language models like the BERT [6], the DistiBERT [9], the RoBERTa [10], and the ParsBERT [7].

2) Many projects try to improve the models by examining several experiments under different conditions. In [11], the authors have described the TransferTransfo; this model combines Transfer learning and the new Transformer language model provided by OpenAI's paper [12]. OpenAI is a language model like BERT, which is based on transformers. In [13], the authors have proposed a new recipe for building a chatbot with different parameters; this work has been converted into a new framework that is able to apply to diverse tasks. In [14], PLATO-2 is introduced via curriculum learning; this model has two stages. The first stage is response generation, called coarse-grained generation, and the second stage is related to generating diverse responses and selecting the best one.

3) Several projects focus on applying chatbots to a specific topic. In [15], the new application was designed to help cryptocurrency

[DOI: 10.61186/itrc.15.2.49]

investors. For depression purposes, [16] has proposed a new chatbot that can understand the user's emotional states and generate a unique response according to the desired goal; this model is based on the LSTM model. The work in [17] which is also trained by the LSTM model, detects customer requests on social media. This application is acceptable in both human and metric evaluations.

In comparison with [15], our chatbot can analyze and respond to more bourse-specefic questions. In addition, this is the first Persian stock-market chatbot to the best of our knowledge. Also, in this project, we represent more metrics via testing the application.

III. THE PROPOSED APPROACH

Our proposed approach is based on the Persian version of BERT language model. To better explain this algorithm, we will start by describing the basic model which have led to BERT.

A. The basic model based on BERT

The RNN model is the basic structure of the chatbot models. The weights of this network are shared in the whole model; also, historical information was saved due to model architecture. We could compute the Activation value in time-step <t> by (1) and the output with (2) [2]:

$$a^{<t>} = g^1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad (1)$$

$$y^{<t>} = g^2(W_{ya}a^{<t>} + b_y) \quad (2)$$

where $x^{<t>}$ and $y^{<t>}$ are respectively the input and the output in time-step <t>, $a^{<t>}$ is the activation value is time-step <t>, $[W_{aa}\,.\,W_{ax}\,,W_{ya}]$ are shared weights, and $[b_a\,.b_y]$ are shared biases. Also, $[g^1\,.g^2]$ are activation functions. Equation (1) shows that the activation value in time-step <t-1> keeps information from the prior layers. This information is applied to the model by $a^{<t>}$; this process is extended until the final output of the sequence is obtained.

One crucial problem of the RNN model is the vanishing/exploding gradient. This issue exponentially decreases/increases with respect to the number of layers. The slope of the gradient plot is related to derivatives in every layer during the backpropagation step. We could address exploding gradient by use of the method called gradient clipping [18], which prevents gradient to increase from a specific value. To address the exploding gradient more effectively and resolve the vanishing gradient issue, we use specific gates in the RNN model. Two models called the LSTM[3] and GRU[3] are obtained according to these gates. Four types of gates are defined: the Update, Relevance, Forget and Output gates. The Update and Relevance gates are shown with Γ_u and Γ_r, respectively, which are used in both LSTM and GRU algorithms, but the Forget and Output gates ($\Gamma_f\,.\Gamma_o$) are exclusive to the LSTM model.

We can compute the four mentioned gates as (3), (4), (5), and (6) [3]:

$$\Gamma_u = \sigma(W_u[a^{<t-1>}.x^{<t>}] + b_c) \quad (3)$$

$$\Gamma_r = \sigma(W_r[a^{<t-1>}.x^{<t>}] + b_r) \quad (4)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}.x^{<t>}] + b_f) \quad (5)$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}.x^{<t>}] + b_o) \quad (6)$$

In equations (3), (4), (5), (6), σ is the activation function and $[W_u, W_r, W_f, W_o]\,, [b_u, b_r, b_f, b_o]$ are the weights and biases related to gates, respectively.

The GRU relationships are computed as (7) and (8) [3]:

$$\tilde{c}^{<t>} = tanh\,(W_c[\Gamma_r * c^{<t-1>}.x^{<t>}] + b_c) \quad (7)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}) \quad (8)$$

where the $c^{<t>}$ value called cell memory is equal to the activation value in time-step <t>, so:

$$c^{<t>} = a^{<t>} \quad (9)$$

The cell memory and activation values are obtained as (10), (11), and (12) [3]:

$$\tilde{c}^{<t>} = tanh\,(W_c[\Gamma_r * c^{<t-1>}.x^{<t>}] + b_c \quad (10)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>} \quad (11)$$

$$a^{<t>} = \Gamma_o * tanh\,(c^{<t>}) \quad (12)$$

Finally, the output of both algorithms is computed as (13) [3]:

$$\tilde{c}^{<t>} = Softmax(W_y a^{<t>} + b_y) \quad (13)$$

Input representation is a substantial part of the training process, which could improve the model. The primary representation method, One-Hot Encoding, could not consider the features for each token in the algorithm. Because of this, Word Embedding techniques are used to represent the sequence in language models like BERT. The embedding of each token is computed by matrix multiplication between one-hot vector and the embedding matrix as (14) [4]:

$$e_j = E * O_j \quad (14)$$

One of the leading algorithms in the field of NLP is the model based on Attention block. In this structure, the RNN blocks attend to essential parts of a sequence. The Attention function is obtained as (15) and (16) [5]:

$$A(q.k.v) = \sum_i^{Tx} \frac{exp\,(q \cdot k^{<j>})}{\sum_j exp\,(q \cdot k^{<j>})} v^{<i>} \quad (15)$$

$$A(Q.K.V) = softmax\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \quad (16)$$

where the q, k, and v are the queries, keys, and values vector for each token respectively.Tx is the length of the input sequence, and Q, K, and V are the matrices that are the set of queries, keys, and values, respectively. The d_k is the dimension of the key values which is added to the formula (15). Specifically, we sum over all tokens of the input sequence and $\exp(q \cdot k^{<j>})$ to compute the softmax and attention function.

Instead of a single attention block, the concatenation of blocks called Head is used in language models. Each Head is the attention function of the three arguments which are composed of the multiplication of Q, K, and V with their related parameter matrices. The concatenation of heads which is called Multi-Head

[DOI: 10.61186/itrc.15.2.49]

attention and each head is computed as (17) and (18) respectively [5]:

$$MultiHead(Q.K.V) = concat(head_1 \dots head_n)Wo \quad (17)$$

$$head_i = Attention\left(QW_i^Q + KW_i^K + VW_i^V\right) \quad (18)$$

where the $[Wo.W_i^Q.W_i^K.W_i^V]$ are the parameter matrices.

Positional Encoding (PE) is an essential part of each language model. Since the attention model could not distinguish the location of the tokens in the sequence, this mechanism is used to describe the order of the tokens in the sequence. Sine and cosine functions are used in different frequencies to compute PEs as in (19), (20) [5]:

$$PE(pos.2i) = sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \quad (19)$$

$$PE(pos.2i+1) = cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right) \quad (20)$$

where i is the dimension, pos is the position, and d represents the dimensionality of input or output.

B. The proposed structure based on ParsBERT

Transformer is the leading part of the language model consisting of the previously mentioned structures [5]. The transformer architecture has an encoder-decoder design. The encoder part consists of N layers which are stacked. Each layer is composed of Multi-Head attention and Feed Forward models. The input and output are added together for each sub-model, and the Normalization sub-layer applies to this summation. The outcome of this structure is fed to a Multi-Head layer of decoder structure. The decoder part is the same as the encoder one, except it has the Masked Multi-Head attention sub-model at the beginning of the layer. The input of this sub-model is the shift-right output of the encoder layer. The entire structure of the Transformer model is shown in Fig. 1 [5].

The BERT model is the multi-layer bidirectional Transformer based model. The base model of this algorithm consists of 12 Transformers and self-attention heads. The hidden size of the last layer is 768. Aggregation of the Token-embedding, Segment-embedding, and Position-embedding forms the input representation of the BERT model. Segment-embedding detaches two sentences in the sequence, and Token-embedding gives an equivalent embedding vector to each token.

In this paper, we use the ParsBERT [7], which is the Persian version of BERT with the same structure. We propose concatenating a new fully-connected layer and the softmax layer with ParsBERT to obtain the number of classes according to our designed dataset. We use the dropout layer in some experiments to overcome the overfitting problem. Fig. 2 shows The basic structure of the proposed model with the dropout layer after the first fully-connected layer (FC1). Also, we add the second fully-connected layer with the softmax activation function (FC2) to predict the output vector. The FC1 layer has 256 neurons, and FC2 has 17 neurons contributing to 17 output classes.

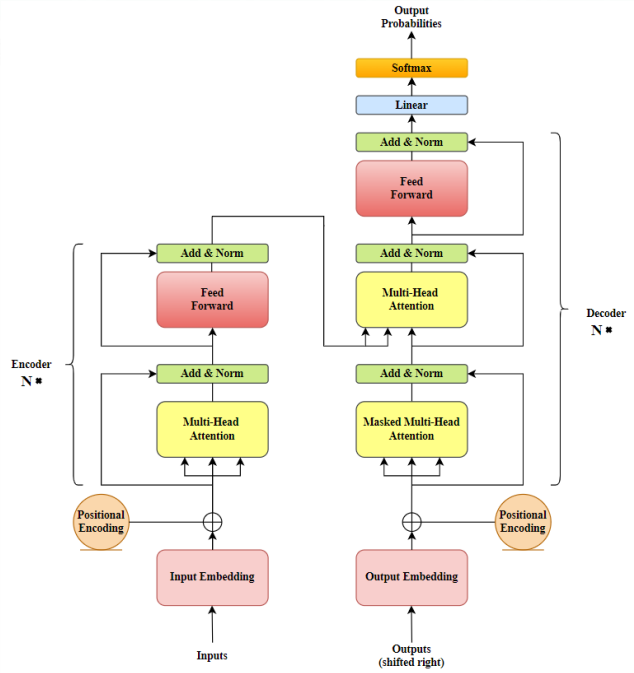


Figure 1. The structure of the Transformer model [5]. The left part of the architecture is the encoder, and the right one is the decoder.

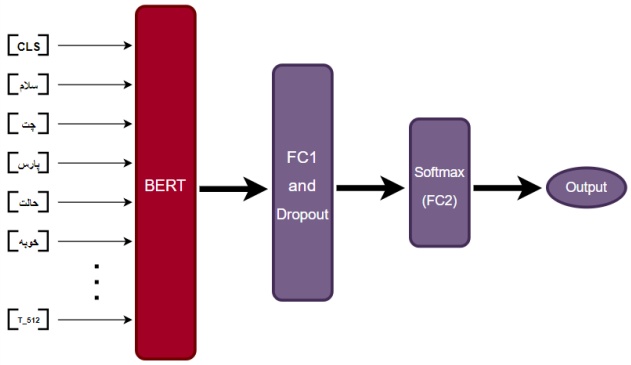


Figure 2. The basic structure of the proposed chatbot model. BERT refers to ParsBERT since we use the Persian dataset.

IV. DATASET

Since there is no proper Persian dataset for the Bourse Chatbot application, we have manually designed the dataset of this project to be appropriate for Bourse Chatbot's application. It consists of 590 questions and 17 classes related to the equivalent answers. Answers are predetermined according to relevant classes. The name and number of categories in this dataset are shown in Table 1. We have selected appropriate categories for financial concepts. These categories are those related to the bourse applications. The questions in each category include both conversational and formal states. Also, we have chosen different lengths for the questions in order to help the model to be generalized. The lengths of designed questions differ from the minimum of 4 characters in the "Greetings" category to the maximum of 91 characters in the "Bourse Menu" class. Also, the average length of all questions in all categories is 37.07 characters. We have determined the answers to be

[DOI: 10.61186/itrc.15.2.49]

much longer to have enough information. The average length of all answers in different categories is 270.24 characters. We could publish our dataset as a public one on GitHub.

We have used the Stratified K-Fold cross-validation with K=4 to split the dataset into train and validation sets. The test set includes the utterances expressed in the application by users. For padding operation, the max-length parameter determines the maximum length of questions. The value of this parameter is 64, which shows that if the question length is more than 64, the additional tokens will be zero. Fig. 3 and Fig. 4 show the distributions of training and validation data regarding sentence length in each fold, respectively.

TABLE I. DISTRIBUTIONS OF CLASSES IN THE DESIGNED DATASET.

Class name in Persian	Class name in English	Counts
منوی بورس	Bourse Menu	46
ریسک	Risk	46
منبع آموزشی	Educational Resource	43
خداحافظی	Goodbye	43
شاخص	Stock Index	40
سیگنال	Signal	39
ثبت نام	Register	38
سهام بنیادی	Basic Share	35
تعاریف سهام	Share Definition	35
اسم	Name	33
احوال پرسی	Greetings	32
سهام تکنیکال	Technical Share	31
شغل	Career	30
صف خرید و فروش	Stock Exchange Queue	28
تعریف بورس	Bourse Definition	26
سن	Age	23
دسته بندی سهام	Share category	21

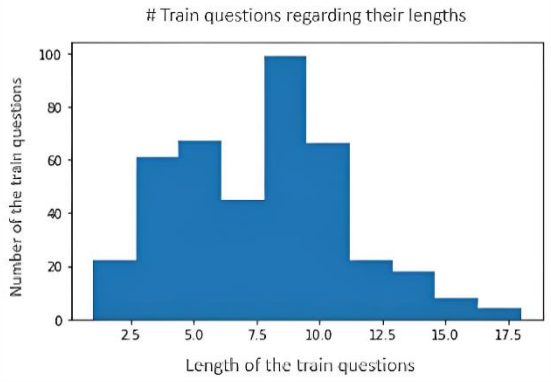


Figure 3. Distributions of the train data regarding their lengths.

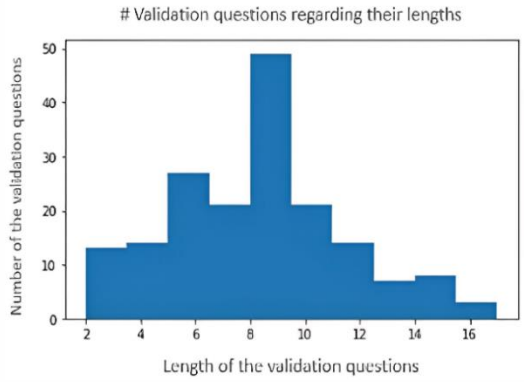


Figure 4. Distributions of the validation data regarding their lengths.

V. EXPERIMENTS

A. Training Procedure

We have selected the following values for the parameters and the hardware:

1) Parameters: We use the Adam optimizer for the optimization process. The batch size is set to 64 to define the number of training examples in each iteration. The number of epochs is 200 which determines how many times the entire train set should be trained. We set the dropout coefficient to 0.3 for the primary model and used the categorical cross-entropy as the loss function. Because of the imbalance distribution of classes, we have used the Stratified K-fold cross-validation with K=4 to divide the dataset into train and validation sets. Then, we have averaged the results of these four folds. This method helps generalize the algorithm to make it unbiased, so the test results will be more reliable. Moreover, we use specific initial weights to balance the dataset for the training process. With the usage of the compute_class_weight function in the Scikit_learn library, we initialize the weights as (21):

$$Wi = n\_samples \,/\, (n\_classes \,/\, np.bincount(y)) \quad (21)$$

where the n_samples parameter shows the number of samples in the dataset, n_classes shows the number of classes, and the np.bincount(y) is the Numpy function that computes the occurrence of each element of the output vector (y).

2) Hardware and Frameworks: In this paper, we use the Pytorch framework to design the model and training processes. Also, the Google collab or the Colaboratory platform is performed to write and execute codes with the necessary capability to access the free GPU. The GPU of the Google collab in free usage is Nvidia K80, the GPU memory is 12 G.B./ 16 G.B., and the GPU memory clock is 0.82GHz / 1.59GHz. We use the Flask framework to design the GUI application. With this Python framework, we could do the back-end programming part. We perform Html, CSS, and Javascript jointly with frameworks such as Bootstrap to develop the front-end part of the web application.

B. Results

We have examined seven experiments and saved the weights of the best model to apply in the application. Based on these experiments and user questions, the results in terms of accuracy and loss, confusion matrix, recall, precision, and F1-score as well as test results are presented in this paper.

1) Accuracy and Loss**:** In Table 2, different parameters and the accuracy on the train and validation sets of seven experiments are shown. The best model is chosen based on the maximum validation accuracy.

[DOI: 10.61186/itrc.15.2.49]

Also, we plot the best model in terms of accuracy and loss metrics in Fig. 5. We set the epoch to 200 and also use early-stopping with patience-time of 30 to overcome overfitting problem. In Table 2, the changes of each experiment parameters compared to the basic one are specified. As we show in Fig. 2, the basic model is composed of the ParsBERT language model concatenating with the fully-connected layer (FC1), the dropout layer, and finally the fully-connected layer which uses the softmax activation function (FC2) to find the output. In our experiments, we change parameters such as the epochs (E), and the learning rate (LR). Also in some experiments, we change the dropout layer after the fully-connected layer (DP1) or after the softmax layer (DP2).

According to the obtained accuracy of the validation set in Table 2, we select the fourth experiment model as the best model and predict the user utterances via this model in the designed application. Thus, the best model has two fully-connected layers and one dropout layer after the second FC layer. Also, we set the number of epochs to 500 based on this experiment.

In Table 3 , we have demonstrated the precision, recall, and F1_score for all models specified in Table 2.

2) Confusion matrix: This metric can specify the model's performance in each class. We evaluate the effectiveness of the imbalance dataset by this matrix. In Fig. 6, the confusion matrix of the validation set is shown for the best model. For better vision, we demonstrate the average classification accuracy of the model for each class on the validation set in Table 4 as well.

C. Test results

We gather test results by recording the questions and responses of the users in the web application in Figs. 7 and 8. We call our designed application ChatParse. According to the results, embedding links and variation in the length of the utterances are capabilities of the predetermined responses.

D. Discussion

As mentioned, we select the fourth experiment as the best model according to the obtained results in Table 2. Fig. 6 shows the confusion matrix of the best chosen model for the validation set. According to the average classification results in Table 4, the accuracy of the class “Bourse Menu” is 83%; while it is 54% for the class “Bourse Definition”. This difference in the results of these two classes is due to the difference in the data distribution of these two classes in the train set according to Table 1. In other words, our dataset is an imbalanced one; and thus, it affects the accuracies of different classes especially those with fewer data. We can improve the suggested algorithm by adding more data to our dataset as well as using more different contexts in the questions of various categories. According to Fig. 7. (a), and Fig. 8, our designed application works well on the test set. However, occurring some errors is inevitable as the confusion matrix in Fig. 6 also illustrates. For example, Fig. 7. (b) shows the sample which is predicted incorrectly. The reason for this false prediction is the similarity between this class which is named “Bourse Definition” and the class “Share Defenition” in terms of concept and words. This resemblance is more visible in the classes such as “Basic Share ” and the “Technical share ” according to the confusion matrix in Fig. 6.

We have selected the fourth experiment as the best model based on the highest accuracy results on the validation set. This experiment also led to the best precision results and almost the same results on F1 score compared to the basic model according to Table 3.

Our designed application (ChatParse) is suitable for answering different questions of the users. Since these utterances are unbiased, this app is completely worthy for the test dataset. For example, the users may have different question styles and use some words different from those seen in the training set.

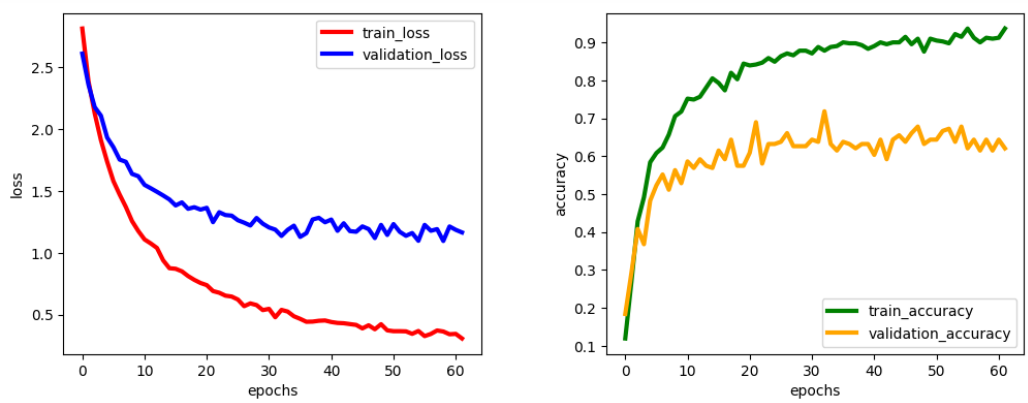
VI. Conclusion and Future Works

In this paper, we designed and developed a brand new Persian stock-market chatbot using the retrieval approach. In this system, we proposed using the Persian version of the BERT called ParsBERT. We also proposed adding a fully-connected layer and softmax layer to the ParsBERT, and examined the model by changing the parameters and number of layers to reach the best model in terms of the loss and accuracy metrics. More importantly, since we could not find any specific Persian data for the Bourse Chatbot application, we manually designed an appropriate dataset for it. Moreover, we have designed and developed the chatbot app called ChatParse which can have multi-turn conversations with users on the stock-market topic. In this paper, we have designed a manual dataset which is still small for an NLP algorithm. Hence, data enhancement in terms of size and context is an effective process which will be considered for future updates. In addition, we could consider the generative models to design a chatbot capable of communicating with users on a general topic.

[DOI: 10.61186/itrc.15.2.49]

TABLE II. THE ACCURACY OF SEVEN EXPERIMENTS ON THE MODEL BY CHANGING DIFFERENT PARAMETERS. FOR EACH EXPERIMENT, L.R. IS THE LEARNING RATE, DP1 IS THE DROPOUT LAYER AFTER THE FIRST FULLY-CONNECTED LAYER, DP2 IS THE DROPOUT LAYER AFTER THE SOFTMAX LAYER (FC2), AND E IS THE EPOCH.

Experiments	Train_accuracy	Validation_accuracy
I. The basic model DP1=0.3 E=200, LR=0.001	97.285	71.429
II. LR=0.00025, Add(DP2=0.1)	89.366	68.027
III. Remove(DP1), Add(DP2=0.1)	93.438	68.707
IV. E=500, Add(DP2=0.1)	94.117	**72.109**
V. DP1=0.4	68.099	53.061
VI. Remove(FC1), Remove(DP1)	**97.737**	70.068
VII. Remove(FC1), Remove(DP1), Add(DP2=0.2)	84.842	63.946



(a) (b)

Figure 5. (a) The loss of the train and the validation set, (b) The accuracy of the train and the validation set.

TABLE III. THE PRECISION, RECALL, AND F1_SCORE VALUES FOR EACH EXPERIMENT ON THE VALIDATION SET.

Experiments	Validation_precision	Validation_recall	Validation_F1Score
I. The basic model DP1=0.3 E=200, LR=0.001	70.8	**72.9**	**71.8**
II. LR=0.00025, Add(DP2=0.1)	71.4	70.3	70.8
III. Remove(DP1), Add(DP2=0.1)	71.8	66.4	69.0
IV. E=500, Add(DP2=0.1)	**72.9**	70.5	71.6
V. DP1=0.4	56.0	43.2	48.7

[DOI: 10.61186/itrc.15.2.49]

VI. Remove(FC1), Remove(DP1)	72.5	69.2	70.6
VII. Remove(FC1), Remove(DP1), Add(DP2=0.2)	71.8	64.3	66.9

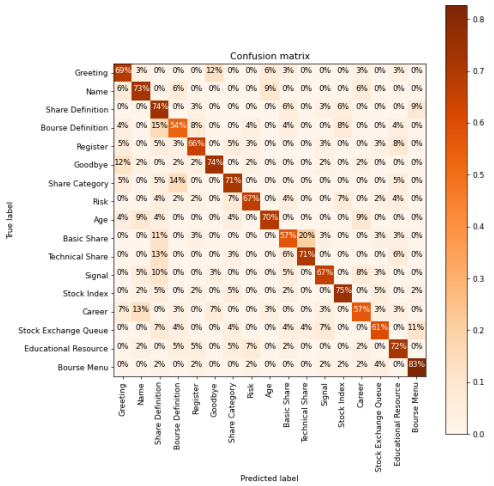


Figure 6. Confusion matrix of the validation dataset for the best proposed model.

TABLE IV. AVERAGE CLASSIFICATION ACCURACY OF THE PROPOSED MODEL FOR EACH CLASS ON THE VALIDATION SET.

Class name in Persian	Class name in English	Accuracy (%)
احوال پرسی	Greeting	69
اسم	Name	73
تعاریف سهام	Share Definition	74
تعریف بورس	Bourse Definition	54
ثبت نام	Register	66
خداحافظی	Goodbye	74
دسته بندی سهام	Share Category	71
ریسک	Risk	67
سن	Age	70
سهام بنیادی	Basic Share	57
سهام تکنیکال	Technical Share	71
سیگنال	Signal	67
شاخص	Stock Index	75
شغل	Career	57
صف خرید و فروش	Stock Exchange Queue	61
منبع آموزشی	Educational Resource	72
منوی بورس	Bourse Menu	83

(a)

(b)

Figure 7. Two samples of test results of the best model in the GUI application. (a): The first and the second questions are related to the "Greetings" and "Goodbye" classes, respectively; (b): the question is related to "Signal" class.

[DOI: 10.61186/itrc.15.2.49]

(a)

(b)

Figure 8. Two samples of test results of the best model in the GUI application. (a): the question is about the “Bourse Menu” class; (b): the first question is about “Greetings” and the second one is about the “Stock Index” class.

REFERENCES

[1] L. Yang, J. Hu, M. Qiu, C. Qu, J. Gao, W.B. Croft, X. Liu, Y. Shen, and J. Liu, “A hybrid retrieval-generation neural conversation model,” arXiv preprint arXiv:1904.09068, 2019.

[2] N. M. Rezk, M. Purnaprajna, T. Nordström and Z. Ul-Abdin, “Recurrent neural networks: an embedded computing perspective,” IEEE Access, vol. 8, pp. 57967-57996, 2020.

[3] B. C. Mateus, M. Mendes, J. T. Farinha, R. Assis, and A. M. Cardoso, “Comparing LSTM and GRU models to predict the condition of a pulp paper press,” Energies, vol. 14, issue 21, 2021.

[4] U. Naseem, I. Razzak, S. Khalid-Khan, and M. Prasad, “A comprehensive survey on word representation models: from classical to state-of-the-art word representation language models,” arXiv preprint arXiv:2010.15036, 2020.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need. arXiv preprint arXiv:1706.03762, 2017.

[6] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.

[7] M. Farahani, M. Gharachorloo, M. Farahani, and M. Manthouri, “ParsBERT: Transformer-based model for Persian language understanding,” Neural Processing Letters, vol. 53, issue 6, pp. 3831–3847, Dec 2021.

[8] M. Schuster, and K. K. Paliwal, “Bidirectional recurrent neural networks,” IEEE Transactions on Signal Processing, vol. 45, issue 11, pp. 2673 – 2681, 1997.

[9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” arXiv preprint arXiv:1910.01108, 2019.

[10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “RoBERTa: A robustly optimized BERT pretraining approach,” arXiv preprint arXiv:1907.11692, 2019.

[11] T. Wolf, V. Sanh, J. Chaumond, C. Delangue, “TransferTransfo: a transfer learning approach for neural network based conversational agents,” arXiv preprint arXiv:1901.08149, 2019.

[12] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” 2018

[13] S. Roller, E. Dinan, N. Goyal, D. Ju, et al. “Recipes for building an open-domain Chatbot,” Proc. Of 16th Conference of the

[DOI: 10.61186/itrc.15.2.49]

European Chapter of the Association for Computational Linguistics, 2021, pp. 300-325.

[14] S. Bao, H. He, F. Wang, H. Wu, H. et al., "PLATO-2: towards building an open-domain Chatbot via curriculum learning," Findings of the Association for Computational Linguistics, 2021, pp. 2513-2525.

[15] Q. Xie, Q. Zhang, D. Tan, T. Zhu, S. Xiao, B. Li, L. Sun, P. Yi, and J. Wang, "Chatbot application on cryptocurrency," IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), 2019.

[16] S. Nagargoje, V. Mamdyal and R. Tapase, "Chatbot for depressed people," United International Journal for Research & Technology (UIJRT), vol. 2, issue 7, pp.208-211, 2021.

[17] A. Xu, Z. Liu, Y. Guo, V. Sinha and R. Akkiraju. "A new Chatbot for customer service on social media." Proc. of the 2017 CHI Conference on Human Factors in Computing Systems, 2017, pp. 3506-3510.

[18] J. Zhang, T. He, S. Sra, A. Jadbabaie, "Why gradient clipping accelerates training: a theoretical justification for adaptivity," International Conference on Learning Representations, 2022.

Ali Shahedi received his B.Sc. degree in Electronics from Amirkabir University of Technology (AUT). He is currently pursuing M.Sc. degree with the Sharif University of Technology (SUT) in device system fields. Before joining the SUT, he worked as the data scientist in the AIMEDIC CO. His interests include Object Detection and Tracking, Data Optimization, and Language Processing.

Sanaz Seyedin received her B.Sc. degree from Amirkabir University of Technology, Tehran, Iran, and the M.Sc. from Iran University of Science and Technology, Tehran, Iran, both in Electronics Engineering, in 2001, and 2005, respectively. She received the Ph.D. degree from Amirkabir University of Technology in 2010 focusing on the field of Speech Recognition. She is a Senior Member (SM) of IEEE. She is currently an Assistant Professor at the Electrical Engineering Department, Amirkabir University of Technology, teaching both undergraduate and graduate courses as well as conducting research in different areas of Machine Learning and AI, Signal Processing (Audio, Speech, Image, Biological Signals), Compressive Sensing and Sparse Coding, and Source Separation.

[DOI: 10.61186/itrc.15.2.49]