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

Persian Ezafeh Recognition using Transformer-Based Models

Ali Ansari*
Computer Engineering Department, Amir-Kabir University of Technology Tehran, Iran
Adak Vira Iranian Rahjoo Company
Tehran, Iran
a.ansari3103@gmail.com

Zahra Ebrahimian
School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran
Adak Vira Iranian Rahjoo Company
Tehran, Iran
z.ebrahimian@ut.ac.ir

Ramin Toosi
School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran
Adak Vira Iranian Rahjoo Company
Tehran, Iran
r.toosi@ut.ac.ir

Mohammad Ali Akhaee
School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran
Adak Vira Iranian Rahjoo Company
Tehran, Iran
akhaee@ut.ac.ir

***Abstract*— In Persian, the grammatical particle ezafe connects two words. Ezafe is one of the salient factors in Persian phonology and morphology to understand the meaning of a sentence completely and truly, whereas it is not usually written in sentences, resulting in mistakes in reading complex sentences and errors in natural language processing tasks. Therefore, recognizing words that need Ezafe at the end of themselves, is a major factor to improve the performance of a variety of NLP-based systems such as a Text TTSsystem. Because in Persian TTS systems without an Ezafe recognition module cannot make Ezafe constructions to read the text correctly and does not recognize the relations between the words. As Transformer-based methods shows state-of-the-art results in lots of NLP tasks, in this paper, we experiment ParsBERT in the task of ezafe recognition. The latter earning 2.68% better F1-score than the prior state-of-the-art, we obtain the most advantageous outcomes.**

Keywords— Ezafe recognition, Natural Language Processing, BERT

I. INTRODUCTION

Ezafe, also known as "Kasreh," is a short, unstressed vowel that is used at the end of words in Persian to connect a head noun, pronoun, adjective, preposition, or adverb to its modifiers in a constituent. It is pronounced as /-e/ after constants and as /-ye/ after vowels. Some common uses of the Persian Ezafe are:

- A noun before an adjective: aseman -e- ziba “beautiful sky”
- A noun before a possessor: ketab -e- man: “my book”
- Some prepositions before nouns: posht -e- dar “behind door”

Ezafe is also found in other languages such as Urdu [1], Kurdish [2], Turkish [3] and Arabic [4]. Recognizing the Persian Ezafe is crucial for understanding and communicating effectively in the Persian language. The Ezafe is a grammatical feature in Persian that functions similarly to the English preposition “of” or the possessive “s” ending. It is represented by the sound “e” or “ye” in written form and is used to indicate a relationship of possession, attribution, or association between two nouns. One of the main reasons why recognizing the Ezafe is important is that it affects the meaning of the sentence. The placement of the Ezafe can change the meaning of a sentence completely. For example, “kitaab-e dost” means “friend’s book,” while “doste kitaab” means “book of a friend.” In the first sentence, the emphasis is on the book, while in the second sentence, the emphasis is on the friend. Additionally, the Ezafe is important for grammar and sentence structure. Persian is a highly inflected language, and the use of the Ezafe is one way in which nouns and adjectives are modified to show their relationship to other words in a sentence. Without the Ezafe, sentences would be more ambiguous, and it would be more difficult to distinguish between subjects and objects. Recognizing the Ezafe is important for written communication in Persian. The Ezafe is an integral part of Persian script, and it is used in almost all written texts, including literature, poetry, and newspapers. If one does not understand the Ezafe, it can be difficult to read and comprehend Persian writing.

Based on Bijankhan corpus [17], Ezafe is used at the end of about 20 percent of the words in Persian sentences which shows the importance of this marker. Therefore, recognizing words that need Ezafe at the end of themselves, is a major factor to improve the performance of a variety of NLP-based systems, too. A TTS system without an Ezafe recognition module cannot make Ezafe constructions to read the text correctly and does not recognize the relations between the

words [5]. Another task that an Ezafe recognition module could come in great help is part-of-speech (POS) tagging. Studies show that having knowledge about Ezafe improves the results of POS-tagging [6]. Machine translation [7], tokenization [8], phrase segmentation, and the head word of a phrase detention [9] are examples of the tasks that Ezafe recognition could improve the overall performance.

II. Related Works

In Ezafe recognition task, various studies have been pursued and they have reached promising results. A variety of methods such as hybrid, rule-based, statistical and transformer networks-based methods are employed for this task. A Persian morphological analyzer was created by Megerdoomian et al. using a rule-based approach [10]. Based on the next words in a phrase, they established an Ezafe characteristic to identify the existence or absence of Ezafe for each word.

In addition, Bijankhan recognised Ezafe using a pattern matching technique. To get a statistical perspective of Ezafe markers, he employed POS tags and semantic labels (such place, time, and ordinal numbers). The 80 most common patterns, including Noun-Noun and Noun-Adjective, were carefully determined [11]. A 10 million corpus was used to extract the most common pairings. In the study carried out by Isapour, the phrase boundaries were determined by analysing the sentences using a Probabilistic Context Free Grammar (PCFG) [12]. The head and modifiers in each sentence are then identified using the retrieved parse tree. The accuracy of Ezafe marker tagging was improved in the final phase by using a rule-based method. 1000 sentences from the Bijankhan corpus were chosen to test the method. The results cannot be expanded for a wider corpus or other applications due to the small number of sentences. They were 93.29% accurate [12].

Ezafe is regarded by Muller and Ghayoomi as a component of the head-driven phrase structure grammar (HPSG) that has been used to formalise Persian syntax and establish phrase boundaries [13]. In order to disambiguate words phonologically and semantically, Nojoumian has developed a Persian lexical discredit which uses finite-state transducers (FST) to insert short vowels into words in sentences. A rule-based method based on the context and POS tags of the words before Ezafe was used to insert it [14]. Koochari predicts the presence or absence of the Ezafe marker using statistics and classification and regression trees. They employ data including Persian morphological qualities, the POS tags of the current word, two words preceding it, and three words following it to train the model. They have 70,000 words in their train set, however there are only 30,382 words in the test corpus. They assess the effectiveness of their strategy using the Kappa factor, which is 98.25% for negative terms and 88.85% for words containing Ezafe [15].

Maximum entropy (ME) and conditional random fields (CRF) techniques are used by Asghari et al. Using the Bijankhan corpus, they were able to achieve 97.21% accuracy for the ME tagger and 97.83% accuracy for the CRF model with a five-window size. [16].

In order to attain a greater accuracy of 98.04% with CRF, they additionally use five Persian-specific characteristics in a hybrid setup with the ME and CRF approaches. Both a rule-based approach and a genetic algorithm are applied in Noferesti and Shamsfard's work [18]. To find words that include Ezafe, they first apply 53 syntactic, morphological, and lexical rules. The genetic algorithm is then used to identify words that contain Ezafe but weren't identified in the earlier step. They used 2.5 million words from the Bijankhan corpus [9] to train and test the algorithm, and they were 95.26 percent accurate. Several sequence labelling techniques, such as CRF1, CRF2, BLSTM, BLSTM+CNN, BERT, and XLMRoBERTa, were utilised by Doostmohammadi [19]. The XLMRoBERTa model's performance, which earned 97.91% precision, 98.37% recall, and 98.14% F1-score, was their best.

III. Proposed Method

As shown in [19], transformer-based methods, BERT and XLMRoBERTa, achieve the state-of-the-art results in the task of ezafe recognition. As a result, inspired by their work, Pars-BERT network trained on Persian language along with some additional pre-processing and post-processing is proposed in this paper. In the continuation of this section, we go through the details of our recommended approach for Ezafe predicting, which consists of the following four steps: 1) dataset selection; 2) data preprocessing; 3) a transformer model; and 4) output post-processing.

A. Dataset

Our suggested method uses the Bijankhan dataset [17], a tagged corpus suitable for natural language processing (NLP) research on the Persian language. The daily news and popular literature were used to compile this collection. This collection's articles have all been categorised into more than 4300 different subject categories, such as political, cultural, and others. The corpus contains 2.6 million hand labelled words and 550 Persian part-of-speech tags. From the 1.7-million-word Bijankhan corpus, we chose 70000 sentences. Because adjacent sentences can be quotes from the same literature, sentences are chosen at random. This corpus, which includes diverse themes like news stories, literary works, scientific textbooks, and casual conversations, makes it a good fit for the suggested strategy. We used 10% of the corpus as test, 10% as validation and the remaining 80% as training data. Table I shows the exact number of word and sentences and Ezafe labels of dataset in each set.

B. Pre-process

The proposed method consists of a pre-processing step to make the data ready to be fed to the classifier network, which are describe bellow.

- Correct Ezafe labels manually: After reviewing primary outputs, we discovered noticeable number of errors in Ezafe labels in the dataset which forced us to correct these labels manually. Words such as "chera" (why) or "inja" (here) were labeled as Ezafe which were incorrect and we corrected them manually.

- Break down long sentences: We use POS tagging to break down long sentences. Employing POS tagging, verbs are detected in a sentence, and then the sentence is divided into short ones that have just one verb. We used the Parsivar POS tagger model for this step. After that, the number of sentences increased from 7011 to 12559.
- Adding comma in sentences: Due to the fact that our selected dataset includes informal dialogues, after a glance to the results, we realized network will be have problem with recognizing Ezafe for words before the commas which are not written in the text. Therefore, the need was felt that before training the main network, unwritten commas in the text should be detected and added to the text before feeding to the proposed classifier. The Pars-Bert model is employed for this task and is trained on sentences that include commas from a 10 million sentence corpus that was collected from Wikipedia. For training our model, we deleted all commas and assigned 1 to words before commas and 0 to other words.

C. Transformer Model

In this section, an overview of Sequence-to-Sequence BERT is provided as a transformer model for Ezafe recognition. In a 2017 study, the transformer neural network was initially developed to address some of the drawbacks of a straightforward RNN [23]. Transformer networks are a type of deep neural network architecture that have gained significant attention in the field of natural language processing (NLP) in recent years [20]. Unlike traditional NLP models that rely on recurrent neural networks or convolutional neural networks, transformer networks do not use any sequential processing and can handle the entire sequence of input at once, making them highly efficient for tasks such as machine translation and language modeling. The transformer architecture is composed of an encoder and a decoder, with attention mechanisms used to enable the network to focus on the relevant parts of the input for a given task. Transformer networks have been shown to achieve state-of-the-art performance on a range of NLP tasks [20], and their success has also inspired applications in other domains such as computer vision [24] and speech processing [25].

One of the most popular transformer networks which is used in many state-of-the-art studies is the BERT (Bidirectional Encoder Representations from Transformers) model [21]. BERT is a transformer-based language model that has been pre-trained on large amounts of text data, and has achieved state-of-the-art results on various natural language processing tasks. BERT was originally developed for English language processing, but has since been adapted and fine-tuned for other languages. BERT uses a masked language modeling objective to pre-train its layers, which involves randomly masking tokens in the input and training the model to predict the missing word based on the surrounding context.

Pars-Bert, on the other hand, is specifically designed for the Persian language and has been trained on a large corpus of Persian text with more than 3.9M documents, 73M sentences, and 1.3B words. In particular, Pars-Bert uses 12 hidden layers, 12 attention heads, 768 hidden sizes. The total number of parameters in this configuration is 110M. For model optimization, they used Adam optimizer with $\beta 1 = 0.9$ and $\beta 2 = 0.98$ is used for 1.9M steps [22]. Pars-Bert has also been pre-trained using a masked language modeling objective, and has been fine-tuned for a range of downstream tasks, such as named entity recognition, sentiment analysis, and machine translation. Pars-Bert exactly like BERT, have demonstrated significant improvements in performance on a range of NLP tasks in Persian language, and their success has inspired us to use Pars-Bert in this research.

We labeled dataset words with 2 labels: positive-gen and negative-gen, then mapped them to 1 and 0 for our model. The longest sentence in our training set has 102 words and we decided to make arrays with 128 lengths. Finally, we fine-tune the Pars-Bert transformer on our prepared dataset using a binary classification objective for 40 epochs. Fig.1 shows a simple diagram of our proposed model.

D. Post-process

Due to our dataset which has a variety of informal and formal Persian language, there was some mistakes that influenced our results at the end, so we decided to correct some of our results with a rule-based method after it was predicted by our model. The most common error that we discovered in our results was false Ezafe recognition for the words which have half-padding at the end. It caused by different

TABLE I. THE NUMBER OF SENTENCES AND WORDS IN THE DATASET

Datasets	Number of		
	Sentences	*Words*	*Positive words*
Train	56082	1381915	265930
Validation	7010	172670	33757
Test	7011	170773	33839
Total	70103	1725358	333526

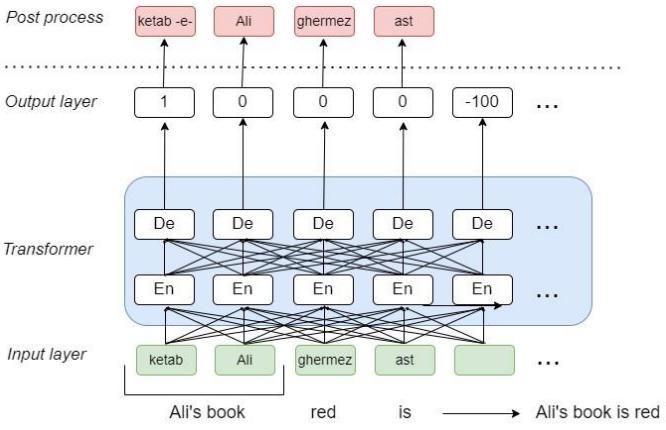

Fig. 1. Model diagram

types of writing half-padding in our dataset. Usually in Persian language, half-padding is used for plural form of words. So, we found plural words and corrected them.

IV. Experiments

In this section, we go over the experimental components of our proposed method for Persian Ezafeh recognition, which include experimental setups to discuss hyper-parameter settings, overall performance to demonstrate our model's results, and performance comparison to compare previous state-of-the-art models with our proposed model.

A. Experimental Setups

To train and evaluate our Persian Ezafeh recognition model, we split the dataset into training (80%), validation (10%), and test (10%) sets. We experimented with different hyper-parameters, including the loss function, optimizer, learning rate, and batch size, to achieve optimal performance. The learning rate was set to 6 e^{-6}, weight decay was set to 0.2, and the batch size was set to 32. For model optimization, we used the Adam optimizer with $\beta1$=0.9 and $\beta2$=0.98 to minimize the binary cross-entropy loss function. We fine-tuned the model for 40 epochs and used early stopping to prevent over-fitting. We implemented the model in Python using the PyTorch library on a workstation with an Intel Core i-7 processor, an NVIDIA GeForce 2080 GPU, and 12GB of RAM.

B. Overall Performance

In this section we demonstrate our model's results. To evaluate the performance of our model, we used precision, recall, and F1score metrics, which are describe bellow.

- Precision: Precision is a measure of how many of the positive predictions made are correct (true positives). It could be calculated as (1). Where TP stands for true positive, FP stands for false positive.

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

- Recall: Recall is a measure of how many of the positive cases the classifier correctly predicted, over all the positive cases in the data. It is sometimes also referred to as sensitivity. The formula for it is shown in (2), where FN stands for False Negative.

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

- F1-Score: F1-Score is a measure combining both precision and recall. It is generally described as the harmonic mean of the two. Harmonic mean is just another way to calculate an "average" of values, generally described as more suitable for ratios (such as precision and recall) than the traditional arithmetic mean. The formula used for F1score in (3).

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (3)$$

Fig. 2 demonstrates evaluation loss of the model through 40 epochs. Evaluation loss was stopped after thirty-fifth epoch near 0.013. Our model achieved an overall F1-score of 99.09% on the test set, demonstrating its high accuracy in recognizing Ezafeh in Persian text. The precision and recall values for our model were 98.87% and 99.09%, respectively.

In order to calculate the impact of pre-processing and postprocessing performed in our proposed method, the results of the proposed network alone, together with pre- and post-processing, are shown in Table II. As can be seen in this table, preprocessing and post-processing had a huge impact on our results and improved our accuracy in recognizing Ezafe. The results show that with the same setups, again we trained our model with the pre-process f1-score increasing from 95.73% to 97.64% and the post-process f1-score improving from 97.64% to 98.98%.

C. Performance Comparison

Due to previous studies, transformer networks outperform the other models by huge margin [19]. So, we compared the performance of our model to this state-of-the-art model for Ezafeh recognition with the same dataset. We have compared our proposed method with XLM-RoBERTa [26] and BERT-multilingual [27] models.

XLM-RoBERTa is a pre-trained language model developed by Facebook AI Research (FAIR) that is based on the RoBERTa model architecture [26]. XLM-RoBERTa stands for Cross-lingual Language Model RoBERTa and it is designed to understand and generate text in multiple languages. The XLM-RoBERTa model is trained on large amounts of text data in multiple languages, using a self-supervised learning approach. During training, the model learns to predict missing words in a sentence or to identify whether two sentences are semantically related. This training approach allows the model to develop a deep understanding of language and the relationships between words and phrases in multiple languages. XLM-RoBERTa has been shown to achieve state-of-the-art performance on a wide range of natural language processing tasks, including text classification, sentiment analysis, and machine translation [27]. Also, Doostmohammadi [19] shows that XLM-RoBERTa achieves state-of-the-art results in ezafeh recognition. The training time of XLM-RoBERTa network on our pre-processed dataset was 7 hours and its evaluation loss plot during the train is shown in Fig.3.

BERT-multilingual is a variant of BERT that is trained on multiple languages [28]. It can handle multiple languages in a single model, making it useful for multilingual applications. BERT-multilingual is trained on a large corpus of text from 104 languages, including English, Spanish, French, Chinese, Arabic, and many others. The advantage of BERT-multilingual is that it can perform well on many different languages, which

is useful for applications that need to process text in multiple languages. However, it is important to note that BERT-multilingual may not perform as well on specific languages as models trained specifically on those languages. Additionally, BERT-multilingual has a larger model size and requires more computational resources than models trained on a single

language. The training time of BERT-multilingual network on our pre-processed dataset was 18 hours and its evaluation loss plot during the train is shown in Fig.4.

Table III illustrates precision, recall, and F1-score on the test set for XLM-RoBERTa, BERT-multilingual, and our proposed method. As can be seen in this table, XLM-RoBERTa outperforms 95.15% precision, 93.64% recall, and 94.39% for the f1-score; likewise, BERT-multilingual outperforms 98.19% precision, 97.82% recall, and 98.00% for the f1-score, respectively. Our model outperformed all previous models, achieving an F1score of 98.98%, precision of 98.87% and recall of 99.09%.

To demonstrating and visualizing results, we compare models' Precision, Recall and F1-score using a bar chart respectively from top to bottom in Fig.5.

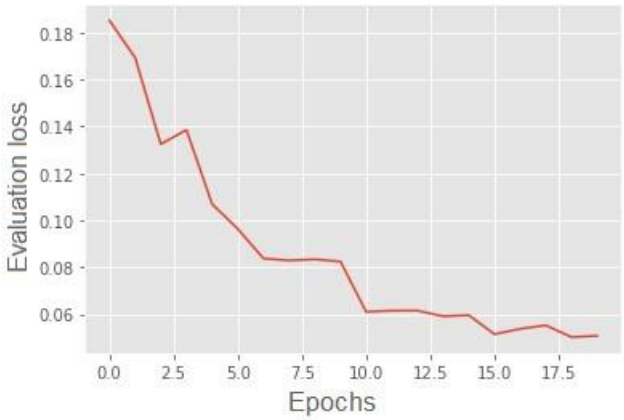

Fig. 2. ParsBert evaluation loss

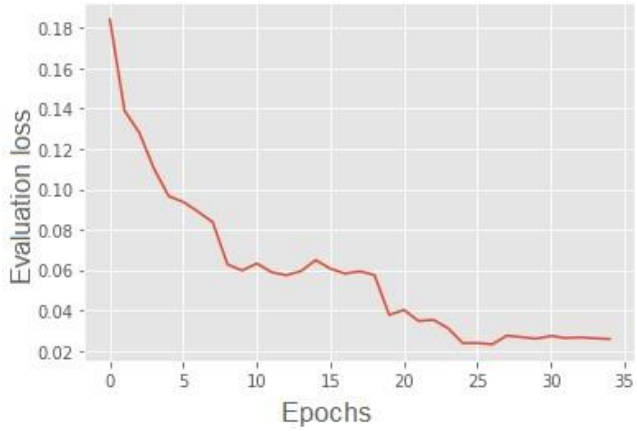

Fig. 3. XLM-RoBERTa evaluation loss

TABLE II. MODEL'S PERFORMANCE BEFORE AND AFTER SETUP PRE-PROCESS AND POST-PROCESS

Models	Percentage of		
	Precision	*Recall*	*F1-score*
Initial modle	95.97%	95.49%	95.73%
After pre-process	96.45%	98.86%	97.64%
After post-process	99.45%	98.51%	98.98%

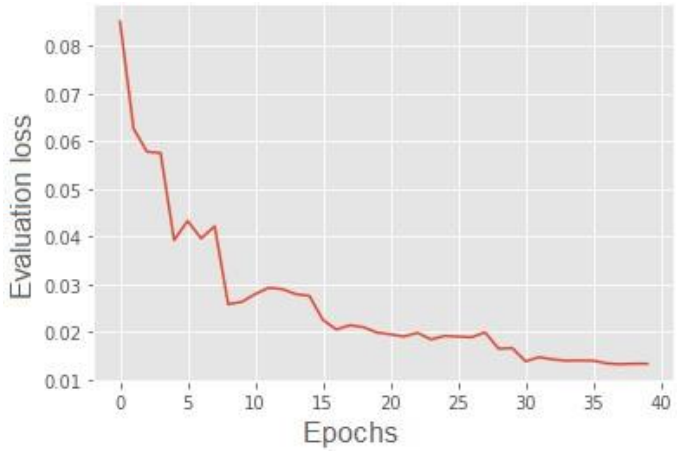

Fig. 4. BERT-multilingual evaluation loss

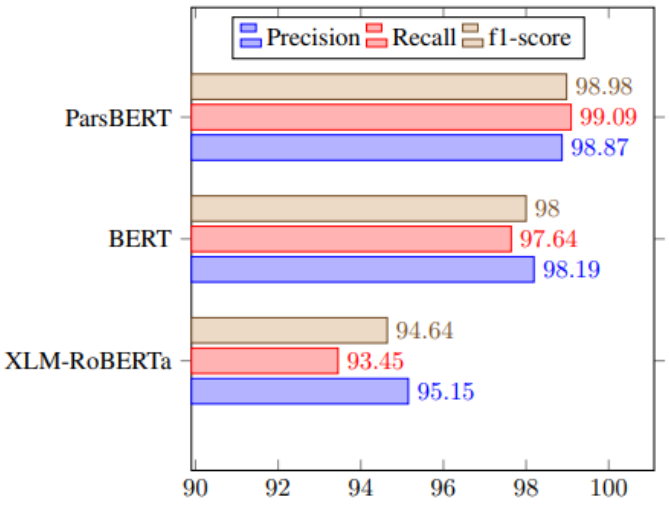

Fig. 5. Ezafe recognition precision, recall, and F1-score, respectively from top to bottom, for all of the models on the test set.

TABLE III. EZAFE RECOGNITION RESULTS

Table Head	Percentage of		
	Precision	*Recall*	*F1-score*
ParsBert	98.87%	99.09%	98.98%
BERT-multilingual	98.19%	97.64%	98%
XLM-RoBERTA	95.15%	93.45%	94.64%

PRECISION, RECALL, F1-SCORE, AND ACCURACY RESPECTIVELY FROM TOP TO BOTTOM, FOR ALL OF THE MODELS ON THE TEST SET.

V. Conclusion and Future Works

In this paper, we propose our approach for Persian Ezafeh recognition using the ParsBert transformer and the Bijankhan dataset with 1.8 million words in seventy thousand sentences. Our experimental results demonstrate the effectiveness of the proposed method and highlight the potential of transformer-based models for complex NLP tasks such as Ezafeh recognition. We have also compared our proposed method with the XLMRoBERTa and BERT multilingual models, which provided the best results in the previous studies, to compare our results with theirs. Our experimental results show that the proposed method outperforms the existing state-of-the-art methods, achieving an F1-score of 98.98%. An exciting direction for future work could be developing new POS tagging models trained with Ezafe marks to reach higher accuracy. Also, using spontaneously rule-based methods and transform models could improve the accuracy of Ezafe's recognition.

Acknowledgment

The Authors would like to especially thank the Adak Vira Iranian Rahjo (Avir) company for providing the essential gear, particularly the GPU, for this study.

References

[1] T. Bogel, M. Butt, and S. Sulger, "Urdu ezafe and the morphology syntax¨ interface," Proceedings of LFG08, 2008.

[2] A. Holmberg and D. Odden, "The Izafe and NP structure in Hawrami," Durham Working Papers in Linguistics, 2004.

[3] G. Van Schaaik, "The noun in Turkish: Its argument structure and the compounding straitjacket: Otto Harrassowitz Verlag," 2002.

[4] Nizar Y Habash, "Introduction to Arabic natural language processing," Morgan & Claypool Publishers. 2010.

[5] M. Sheikhan, M. Tebyani, and M. Lotfizad, "Continuous speech recognition and syntactic processing in Iranian Farsi language," International Journal of Speech Technology, vol. 1, pp. 135-141, 1997.

[6] Zahra Hosseini Pozveh, Amirhassan Monadjemi, Ali Ahmadi, "Persian texts part of speech tagging using artificial neural networks," Journal of Computing and Security, 3(4):233–241. 2016.

[7] J. W. Amtrup, H. M. Rad, K. Megerdoomian, R. Zajac, "PersianEnglish machine translation: An overview of the Shiraz project," Citeseer, 2000.

[8] M. Ghayoomi and S. Momtazi, "Challenges in developing Persian corpora from online resources," in Asian Language Processing, IALP'09," International Conference on, 2009, pp. 108-113. 2009.

[9] P. Samvelian, "The Ezafe as a head-marking inflectional affix: Evidence from Persian and Kurmanji Kurdish," Aspects of Iranian Linguistics: Papers in Honor of Mohammad Reza Bateni, pp. 339-361, 2007.

[10] Karine Megerdoomian et al. "Persian computational morphology: A unification-based approach," Computing Research Laboratory, New Mexico State University. 2000.

[11] Mahmood Bijankhan. "The persian text corpus. In In 1st Workshop on Persian Language and Computer," Tehran. 2004

[12] S Isapour, M Homayounpour, M Bijabkhan, "The prediction of ezafe construction in persian by using probabilistic context free grammar," In In Proceedings of 13th Annual Conference of Computer Society of Iran. 2008.

[13] Stefan Muller and Masood Ghayoomi, "Pergram: A trale implementation of an hpsg fragment of persian. In Proceedings of the International Multi-conference on Computer Science and Information Technology, pages 461–467," IEEE. 2010.

[14] Peyman Nojoumian, "Towards the Development of an Automatic Digitizer for the Persian Orthography based on the Xerox Finite State Transducer.University of Ottawa (Canada)," 2011.

[15] Abbas Koochari, Behrang QasemiZadeh, Mojtaba Kasaeiyan, "Ezafe prediction in phrases of farsi using cart," In Proceedings of the I International Conference on Multidisciplinary Information Sciences and Technologies, pages 329–332. 2006.

[16] Habibollah Asghari, Jalal Maleki, Heshaam Faili. "A probabilistic approach to Persian Ezafe recognition." In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers,pages 138–142. 2014.

[17] Mahmood Bijankhan, Javad Sheykhzadegan, Mohammad Bahrani, and Masood Ghayoomi. "Lessons from building a Persian written corpus: Peykare," Language resources and evaluation,45(2):143–164. 2011.

[18] Samira Noferesti and Mehrnoush Shamsfard. "A hybrid algorithm for recognizing the position of Ezafe constructions in Persian texts," IJIMAI, 2(6):17–25. 2014.

[19] Ehsan Doostmohammadi, Minoo Nassajian, and Adel Rahimi. "Persian Ezafe Recognition Using Transformers and Its Role in Part-Of-Speech Tagging. In Findings of the Association for Computational Linguistics," EMNLP 2020, pages 961–971, Online. Association for Computational Linguistics. 2020.

[20] Adrian M. P., Bras¸oveanu, Razvan Andonie. Visualizing Transformers for¨ NLP: A Brief Survey. IEEE. 2020

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," CoRR, vol. abs/1810.04805, 2018.

[22] Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, Mohammad Manthouri. "ParsBERT: Transformer-based Model for Persian Language Understanding," ArXiv, vol. abs/2005.12515, 2020.

[23] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

[24] Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y. and Yang, Z., 2022. A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence, 45(1), pp.87-110.

[25] Karita, S., Chen, N., Hayashi, T., Hori, T., Inaguma, H., Jiang, Z., Someki, M., Soplin, N.E.Y., Yamamoto, R., Wang, X. and Watanabe, S., 2019, December. A comparative study on transformer vs rnn in speech applications. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 449-456). IEEE.

[26] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzman, F., Grave, E., Ott, M., Zettlemoyer, L. and Stoyanov, V., 2019.´ Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116.

[27] Li, B., He, Y. and Xu, W., 2021. Cross-lingual named entity recognition using parallel corpus: A new approach using xlm-roberta alignment. arXiv preprint arXiv:2101.11112.

[28] Liu, C.L., Hsu, T.Y., Chuang, Y.S. and Lee, H.Y., 2020. What makes multilingual BERT multilingual?. arXiv preprint arXiv:2010.10938.