Persian *Ezafe* Recognition Using Transformers and Its Role in Part-Of-Speech Tagging

Ehsan Doostmohammadi♣**, Minoo Nassajian**♣**, Adel Rahimi**♠

♣Sharif University of Technology, Tehran, Iran
♠Dathena Science Pte. Ltd., Singapore
{e.doostm72,m.nassajian2016}@student.sharif.edu,
adel.rahimi@dathena.io

Abstract

Ezafe is a grammatical particle in some Iranian languages that links two words together. Regardless of the important information it conveys, it is almost always not indicated in Persian script, resulting in mistakes in reading complex sentences and errors in natural language processing tasks. In this paper, we experiment with different machine learning methods to achieve state-of-the-art results in the task of *ezafe* recognition. Transformer-based methods, BERT and XLMRoBERTa, achieve the best results, the latter achieving 2.68% F_1-score more than the previous state-of-the-art. We, moreover, use *ezafe* information to improve Persian part-of-speech tagging results and show that such information will not be useful to transformer-based methods and explain why that might be the case.

1 Introduction

Persian *ezafe* is an unstressed morpheme that appears on the end of the words, as *-e* after consonants and as *-ye*[1] after vowels. This syntactic phenomenon links a head noun, head pronoun, head adjective, head preposition, or head adverb to their modifiers in a constituent called '*ezafe* construction' (Nassajian et al., 2019). Whether a word in a sentence receives or does not receive *ezafe* might affect that sentence's semantic and syntactic structures, as demonstrated in Examples 1a and 1b in Figure 1. There are some constructions in English that can be translated by *ezafe* construction in Persian. For instance, English 'of' has the same role as Persian *ezafe* to show the part-whole relation, the relationship of possession, or ''s' construction, and possessive pronouns followed by nouns showing genitive cases are mirrored by Persian *ezafe* (Karimi and Brame, 2012).

This affix is always pronounced but almost always not written, which results in a high degree of ambiguity in reading and understanding Persian texts. It is hence considered as one of the most interesting issues in Persian linguistics, and it has been discussed in details from phonological aspects (Ghomeshi, 1997), morphological aspects (Samvelian, 2006, 2007) and (Karimi and Brame, 2012), and syntactic aspects (Samiian, 1994; Larson and Yamakido, 2008; Kahnemuyipour, 2006, 2014, 2016).

Nearly 22% of the Persian words have *ezafe* (Bijankhan et al., 2011), which shows the prevalence of this marker. Moreover, this construction also appears in other languages such as Hawramani (Holmberg and Odden, 2005), Zazaki (Larson and Yamakido, 2006; Toosarvandani and van Urk, 2014), Kurdish (Karimi, 2007) etc. *Ezafe* construction is also similar to *idafa* construction in Arabic and construct state in Hebrew (Habash, 2010; Karimi and Brame, 2012) and Zulu (Jones, 2018).

Ezafe recognition is the task of automatically labeling the words ending with *ezafe*, which is crucial for some tasks such as speech synthesis (Sheikhan et al., 1997; Bahaadini et al., 2011), as *ezafe* is always pronounced, but rarely written. Furthermore, as recognizing the positions of this marker in sentences helps determine phrase boundaries, it highly facilitates other natural language processing (NLP) tasks, such as tokenization (Ghayoomi and Momtazi, 2009), syntactic parsing (Sagot and Walther, 2010; Nourian et al., 2015), part-of-speech (POS) tagging (Hosseini Pozveh et al., 2016), and machine translation (Amtrup et al., 2000).

In this paper, we experiment with different methods to achieve state-of-the-art results in the task of *ezafe* recognition. We then use the best of these methods to improve the results for the task of POS tagging. After establishing a baseline for this task, we provide the *ezafe* information to the POS tag-

[1] The *y* is called an intrusive *y* and is an excrescence between two vowels for the ease of pronunciation.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 961–971
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

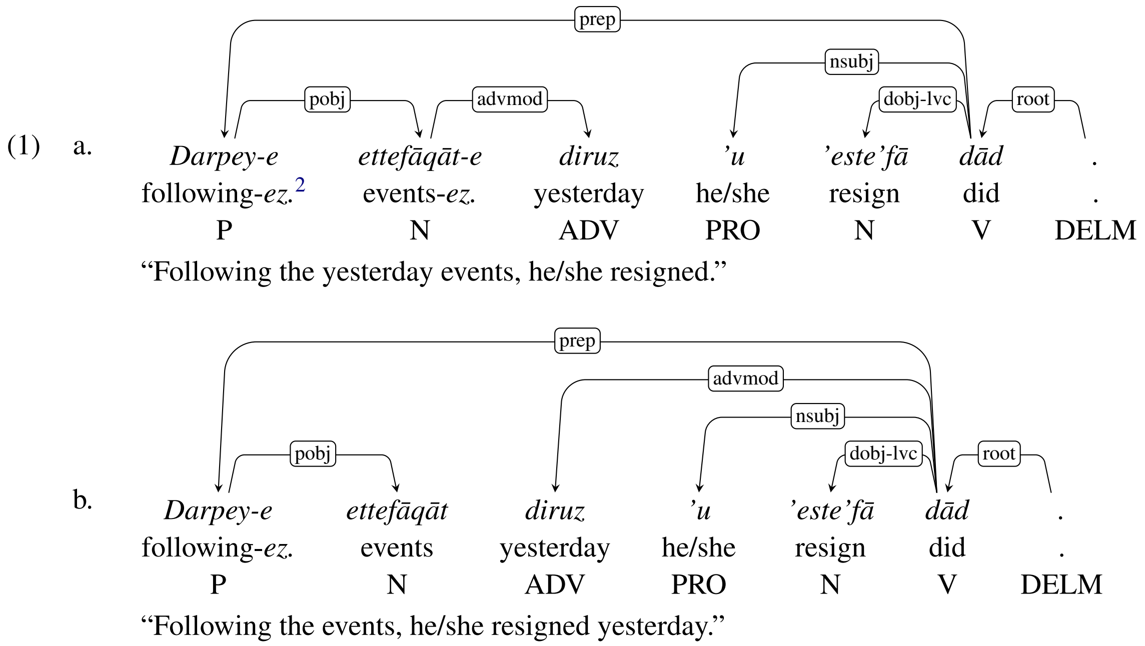
(1) a.
prep
nsubj
pobj
advmod
dobj-lvc
root
Darpey-e ettefāqāt-e diruz 'u 'este'fā dād .
following-ez.[2] events-ez. yesterday he/she resign did .
P N ADV PRO N V DELM
"Following the yesterday events, he/she resigned."
b.
prep
advmod
nsubj
pobj
dobj-lvc
root
Darpey-e ettefāqāt diruz 'u 'este'fā dād .
following-ez. events yesterday he/she resign did .
P N ADV PRO N V DELM
"Following the events, he/she resigned yesterday."

Figure 1: An example of the role of *ezafe* in the syntactic and semantic structures.

ging model once in the input text and the other time as an auxiliary task in a multi-task setting, to see the difference in the results. The contributions of this paper are (1) improving the state-of-the-art results in both of *ezafe* recognition and POS tagging tasks, (2) analyzing the results of *ezafe* recognition task to pave the way for further enhancement in the future work, (3) improving POS tagging results in some of the methods by providing *ezafe* information and explaining why transformer-based models might not benefit from such information. The code for our experiments is available on this project's GitHub repository [3].

After reviewing the previous work of both tasks in Section 2, we introduce our methodology in Section 3 and data in Section 4. We then discuss *ezafe* recognition and POS tagging tasks and their results in Sections 5 and 6, respectively.

2 Previous Work

2.1 *Ezafe* Recognition

In the field of NLP, a few studies have been carried out on Persian *ezafe* recognition, including rule-based methods, statistical methods, and hybrid methods. Most of the previous work on the task rely on long lists of hand-crafted rules and fail to achieve high performance on the task.

Megerdoomian et al. (2000) use a rule-based method to design a Persian morphological analyzer. They define an *ezafe* feature to indicate the presence or absence of *ezafe* for each word based on the following words in a sentence. Another work is Müller and Ghayoomi (2010) that considers *ezafe* as a part of implemented head-driven phrase structure grammar (HPSG) to formalize Persian syntax and determine phrase boundaries. In addition, Nojoumian (2011) designs a Persian lexical diacritizer to insert short vowels within words in sentences using finite-state transducers (FST) to disambiguate words phonologically and semantically. They use a rule-based method to insert *ezafe* based on the context and the POS tags of the previous words.

As for the statistical approach, Koochari et al. (2006) employ classification and regression trees (CART) to predict the absence or presence of *ezafe* marker. They use features such as Persian morphosyntactic characteristics, the POS tags of the current word, two words before, and three words after the current word to train the model. Their train set contains approximately 70,000 words, and the test corpus consists of 30,382 words. To evaluate the performance of the model, they use Kappa factor, and they report 98.25% accuracy in the case of non-*ezafe* words and 88.85% in the case of words with *ezafe*. As another research, we can mention Asghari et al. (2014) that employs maximum entropy (ME) and conditional random fields (CRF) methods. They use the 10 million word Bijankhan

[2] *Ezafe.*

[3] https://github.com/edoost/pert

corpus (Bijankhan et al., 2011) and report an accuracy of 97.21% for the ME tagger and 97.83% for the CRF model with a window of size 5. They also utilize five Persian specific features in a hybrid setting with the models to achieve the highest accuracy of 98.04% with CRF.

Isapour et al. (2008) propose a hybrid method to determine *ezafe* positions using probabilistic context-free grammar (PCFG) and then consider the relations between the heads and their modifiers. The obtained accuracy is 93.29%, reportedly. Another work is Noferesti and Shamsfard (2014) that uses both a rule-based method and a genetic algorithm. At first, they apply 53 syntactic, morphological, and lexical rules to texts to determine words with *ezafe*. Then, the genetic algorithm is employed to recognize words with *ezafe*, which have not been recognized at the previous step. To train and test the model, they use the 2.5 million word Bijankhan corpus (Bijankhan, 2004) and obtain an accuracy of 95.26%.

2.2 POS Tagging

Azimizadeh et al. (2008) use a trigram hidden Markov model trained on the 2.5 million word Bijankhan corpus. In order to evaluate, a variety of contexts such as humor, press reports, history, and romance are collected with 2000 words for each context. The average accuracy on different contexts is 95.11%. Mohseni and Minaei-Bidgoli (2010) also train a trigram Markov tagger on the 10 million word Bijankhan corpus. However, the lemma of each word is determined by a morphological analyzer at first and then a POS tag is assigned to the word. They report an accuracy of 90.2% using 5-fold cross-validation on the corpus. Hosseini Pozveh et al. (2016) use *ezafe* feature for Persian POS tagging. They use the 2.5 million word Bijankhan corpus to train a recurrent neural network-based model, whose input vectors contain the left and the right tags of the current word plus the probability of *ezafe* occurrence in the adjacent words, achieving a precision of 94.7%. Rezai and Mosavi Miangah (2017) design a POS tool based on a rule-based method containing both morphological and syntactic rules. They use the tag set of the 2.5 million word Bijankhan corpus, and their test set is a collection of more than 900 sentences of different types, including medicine, literature, science, etc., and the obtained accuracy is 98.6%. Mohtaj et al. (2018) train two POS taggers on the 2.5 million word Bijankhan corpus, ME and CRF with different window sizes, the best results of which are 95% for both models with a window size of 5.

3 Methodology

We see both *ezafe* recognition and POS tagging as sequence labeling problems, i.e., mapping each input word to the corresponding class space of the task. For the *ezafe* recognition task, the class space size is two, 0 for words without and 1 for words with *ezafe*. The class space size for POS tagging task is 14, consisting of the coarse-grained POS tags in the 10 million word Bijankhan corpus. The results in Section 2.1 are unfortunately reported on different, and in most cases irreproducible, test sets, using accuracy as the performance measure (which is insufficient and unsuitable for the task), making the comparison difficult. We hence re-implemented the model that reports the highest accuracy on the largest test set and compare its results with ours.

3.1 Models

We experiment with three types of models: conditional random fields (CRF) (Lafferty et al., 2001), recurrent neural networks (RNN) (Rumelhart et al., 1986) with long short-term memory (LSTM) cells (Hochreiter and Schmidhuber, 1997) and convolutional neural networks (CNN), and transformer-based (Vaswani et al., 2017) models such as BERT (Devlin et al., 2018) and XLMRoBERTa (Conneau et al., 2019). These are the only transformer-based models pretrained on Persian data. To implement these models, we used sklearn-crfsuite (Korobov, 2015; Okazaki, 2007), TensorFlow (Abadi et al., 2015), PyTorch (Paszke et al., 2019), and HuggingFace's Transformers (Wolf et al., 2019) libraries. The implementation details are as follows:

- CRF$_1$: This is a re-implementation of Asghari et al. (2014)'s CRF model, as described in their paper. The input features were the focus word, 5 previous and 5 following words. We set the L1 and L2 regularization coefficients to 0.1 and the max iteration argument to 100.

- CRF$_2$: This one is the same as CRF$_1$, plus 8 other features: 1 to 3 first and last characters of the focus word to capture the morphological information and two Boolean features indicating if the focus word is the first/last word of the sentence.

- BLSTM: A single layer bidirectional LSTM with a hidden state size of 256 plus a fully-connected network (FCN) for mapping to the class space. The input features were Persian word embedding vectors by FastText (Bojanowski et al., 2017) without subword information with an embedding size of 300, which is proven to yield the highest performance in Persian language (Zahedi et al., 2018). The batch size was set to 16 for *ezafe* recognition and 4 for POS tagging, and learning rate to $1e-3$. We applied a dropout of rate 0.5 on RNN's output and used cross-entropy as the loss function.

- BLSTM+CNN[4]: The same as above, except for the input features of the BLSTM layer, which also included extracted features from dynamic character embeddings of size 32 by two CNN layers with stride 1 and kernel size 2, followed by two max-pooling layers with pool size and stride 2. The first CNN layer had 64 filters and the second one 128. We also applied a dropout of rate 0.5 on CNN's output. The character embeddings were initialized randomly and were trained with other parameters of the model.

- BERT and XLMRoBERTa: The main models plus a fully-connected network mapping to the tag space. The learning rate was set to $2e-5$ and the batch size to 8. As for the pre-trained weights, for BERT, the multilingual cased model and for XLMRoBERTa, the base model were used. We have followed the recommended settings for sequence labeling, which is to calculate loss only on the first part of each tokenized word. Cross entropy was used as the loss function.

We used Adam (Kingma and Ba, 2014) for optimizing all the deep models above. For *ezafe* recognition, we train the models in a single-task setting. For POS tagging, however, we train them in three different settings:

1. A single-task setting without *ezafe* information for all of the models.

2. A single-task setting with *ezafe* information in the input. The outputs of the best *ezafe* recognition model were added to the input of the POS tagging models: for CRFs as a Boolean feature, for BLSTM+CNN as input to CNN, and for BERT and XLMRoBERTa, in the input text. This setting was experimented with using all the models, except for CRF_1 and BLSTM.

3. A multi-task setting where the model learns POS tagging and *ezafe* recognition simultaneously, which means there is an FCN mapping to the POS class space and another one mapping to the *ezafe* class space. For the BLSTM+CNN model, we used a batch size of 16 in this setting. The loss was calculated as the sum of the output losses of the two last fully-connected networks in this setting.

The hyper-parameters of the abovementioned models have been tuned by evaluating on the validation set to get the highest F_1-score. An Intel Xeon 2.30GHz CPU with 4 cores and a Tesla P100 GPU were used to train these models.

3.2 Performance Measure

Precision, recall, F_1-score, and accuracy were used to measure the performance of each model. In all the cases, the model was tested on the test set, using the checkpoint with the best F_1-score on the validation set. For the *ezafe* recognition task, we report the measures on the positive class, and for the POS tagging task, we report the macro average.

4 Data

The 10 million word Bijankhan (Bijankhan et al., 2011) corpus was used in the experiments. We shuffled the corpus, as adjacent sentences might be excerpts from the same texts, with a random seed of 17 using Python's random library. This corpus comprises different topics, including news articles, literary texts, scientific textbooks, informal dialogues, etc, making it a suitable corpus for our work. We used the first 10% of the corpus as the test, the next 10% as validation, and the remaining 80% as the train set. $\sim$22% of the words have *ezafe* marker and $\sim$78% of them do not, in each and all of the sets. Sentences with more than 512 words were set aside. Table 1 shows the number of sentences and tokens in each set.

Table 2 shows the frequency percentage of *ezafe* per POS in the corpus. Despite the previous claim that only nouns, adjectives, and some prepositions

[4] Number of parameters are 3.4M and 9.0M for BLSTM and BLSTM+CNN, respectively.

Set	# of Tokens	# of Sentences
Train	8,079,657	268,740
Valid.	1,011,338	33,592
Test	1,010,274	33,593
Total	10,101,269	335,925

Table 1: The number of sentences and tokens in train, validation, and test sets.

accept *ezafe* (Ghomeshi, 1997; Karimi and Brame, 2012; Kahnemuyipour, 2014), there is actually no simple categorization for POS's that accept *ezafe* and those that do not, which can be seen in Table 2 and is also backed by a more recent study on the matter (Nassajian et al., 2019). The last column in Table 2, H, is Shannon's diversity index (Shannon, 1948; Spellerberg and Fedor, 2003), and is calculated as a diversity measure using Equation 1 for each POS tag. The higher the index is, the more diverse distribution the unique words have.

POS	% w/ *Ezafe*	Freq. %	*H*
N	46.68%	38.50%	8.518
ADJ	24.87%	9.02%	7.468
P	10.10%	10.90%	2.034
DET	9.83%	2.42%	1.944
ADV	5.67%	1.78%	5.289
NUM	2.71%	4.44%	3.573
MISC	1.59%	0.10%	3.735
PRO	1.14%	2.49%	2.884
FW	0.73%	0.22%	7.735
CON	0.12%	9.37%	1.519
V	0.00%	9.58%	5.354
PSTP	0.00%	1.42%	0.029
IDEN	0.00%	0.21%	3.366
DELM	0.00%	9.54%	1.695

Table 2: Frequency percentage of *ezafe* per POS, word frequency percentage per POS, and Shannon's diversity index (H) per POS.

$$H = -\sum_{i=1}^{N} P(x_i) \ln P(x_i) \tag{1}$$

where H is Shannon's diversity index, and N is the number of unique words x in each POS tag.

5 *Ezafe* Recognition

For *ezafe* recognition, we experimented with different sequence labeling techniques and report the performance of them. These techniques include CRF$_1$, CRF$_2$, BLSTM, BLSTM+CNN, BERT, and XLMRoBERTa, as discussed in Section 3.1.

5.1 Results

Table 3 shows the results of all the models on the validation and test sets. It can be seen that transformer-based models outperform the other models by a huge margin. The best RNN-based model, BLSTM+CNN, outperforms the best CRF model, CRF$_2$, by 0.76% F$_1$-score. On the other hand, the best transformer-based model, XLM-RoBERTa, outperforms the best RNN by 1.78% F$_1$-score, and the best CRF by 2.54%. It should be noted that XLMRoBERTa outperforms the previous state-of-the-art, CRF$_1$, by 2.68% F$_1$-score. Figure 2 shows the precision, recall, and F$_1$-score on the test set. The transformer-based models also enjoy a more balanced precision and recall, which means a higher F$_1$-score. It is worth mentioning that XLMRoBERTa has a lower training time due to its much larger pretraining Persian data compared to BERT.

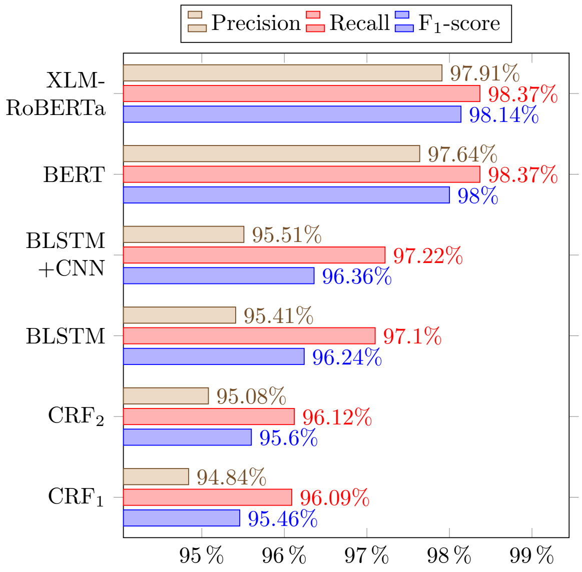

Figure 2: *Ezafe* recognition precision, recall, and F$_1$-score, respectively from top to bottom, for all of the models on the test set.

5.2 Analysis

In comparison to CRFs and RNN-based methods, transformer-based models perform much better on more scarce language forms, such as literary texts and poetry, which means, given a test corpus with a higher frequency of such texts, a much wider gap between the results is expected. We performed an error analysis specifically on XLMRoBERTa's

Model	Validation				Test				Approx.
	Prec.	Recall	F$_1$	Acc.	Prec.	Recall	F$_1$	Acc.	T.T.
CRF$_1$ (baseline)	0.9501	0.9613	0.9556	0.9805	0.9484	0.9609	0.9546	0.9801	0.3 h
CRF$_2$	0.9525	0.9621	0.9573	0.9812	0.9508	0.9612	0.9560	0.9807	0.4 h
BLSTM	0.9541	0.9712	0.9625	0.9880	0.9541	*0.9710*	0.9624	0.9878	0.8 h
BLSTM+CNN	*0.9547*	*0.9721*	*0.9633*	*0.9887*	*0.9551*	<u>0.9722</u>	*0.9636*	*0.9889*	1 h
BERT	<u>0.9767</u>	**0.9839**	<u>0.9803</u>	<u>0.9913</u>	<u>0.9764</u>	**0.9837**	<u>0.9800</u>	<u>0.9912</u>	1.3 h
XLMRoBERTa	**0.9784**	<u>0.9836</u>	**0.9810**	**0.9917**	**0.9791**	**0.9837**	**0.9814**	**0.9919**	0.8 h

Table 3: *Ezafe* recognition results (precision, recall, F$_1$-score, and accuracy) on the validation and test sets. In each column, the best result(s) is/are in bold, the second best underlined, and the third best italicized. The last column shows the approximate training time in hours.

outputs to better understand its performance. We report *ezafe* F$_1$-score per POS tag in order of performance in Table 4.

POS	*Ezafe* F$_1$	POS	*Ezafe* F$_1$
P	99.78%	NUM	92.19%
DET	98.60%	CON	91.16%
N	98.14%	PRO	84.74%
ADJ	96.61%	MISC	53.85%
ADV	95.13%	FW	30.43%

Table 4: *Ezafe* F$_1$-score per POS for XLMRoBERTa's outputs on the test set. The average F$_1$-score is 84.06%.

- Preposition (P): With a relatively low diversity and a high frequency, according to Table 2, prepositions are the easiest one to label for the *ezafe* recognizing model. In addition, prepositions are exclusive in *ezafe* acceptance 93% of the time, making this POS quite easy. The most prevalent error in this POS is the model mistaking the preposition *dar* "in" with the noun *dar* "door", the second of which accepting *ezafe* almost half of the time.

- Determiners (DET): They are easy to recognize partly due to their low diversity. In this POS, the model fails to recognize *ezafe* specifically when the word shares another POS in which it differs in *ezafe* acceptance, e.g., *hadde'aksar* "maximum" and *bištar* "mostly, most of", which accept *ezafe* in DET role, but not in ADV.

- Nouns (N): Despite its high diversity, the model shows high performance in detecting *ezafe* in this POS. This is probably due to its high frequency and high *ezafe* acceptance. Morphological information helps the most in this POS, as many nouns are derived or inflected forms of the existing words. The performance suffers from phrase boundaries detection, which results in false positives. The model also fails to recognize *ezafe* on low-frequency proper nouns, such as Shakespeare. Another common error in this POS is the combination of first and last names, which are usually joined using *ezafe*.

- Adjective (ADJ) and Adverbs (ADV): Both mainly suffer from wrong detection of phrase boundaries, i.e., stopping too early or too late. For instance, look at Example 2 (the error is in bold):

(2) *te'ātr-e* *'emruz-**e*** *qarb*
theater-*ez.* contemp.-*ez.* west
"contemporary western theater"

- Numbers (NUM): The errors in this POS comprise mainly the cardinal numbers, especially when written in digits. The main reason could be the scarcity of digits with *ezafe*. For instance, look at Example 3 (the error is in bold):

(3) *sāl-e* *1990-**e*** *milādi*
year-*ez.* 1990-*ez.* Gregorian
"year 1990 of the Gregorian calendar"

- Conjunctions (CON): It is quite rare for a conjunction to accept *ezafe*, which consequently causes error in *ezafe* recognition.

- Pronouns (PRO): PRO has a low *ezafe* acceptance rate and a low frequency, which makes it a difficult POS. Most of the errors in this POS occur for the emphatic pronoun *xod* "itself, themselves, etc.", which receives *ezafe*, as opposed to its reflective role, which does not.

- Miscellaneous (MISC): Low *ezafe* acceptance and low frequency are the main reasons for the errors in this POS. The errors mainly consist of Latin single letters in scientific texts. Look at Example 4, for instance (the error is in bold):

(4) *L-**e*** *be* *dast* *'āmade*
L-*ez.* to hand come
"the obtained [value of] L"

- Foreign words (FW): With a very low frequency, very low *ezafe* acceptance rate, and a very high diversity, this POS is by far the most difficult one for the model. Additionally, FW usually appears in scientific and technical texts, which makes it harder for the model, as such texts contain a considerable amount of specialized low-frequency vocabulary. Examples of errors in this POS are 'DOS', 'Word', 'TMA', 'off', 'TWTA', etc.

As discussed above, errors are most prevalently caused by model's mistaking phrase boundaries and homographs that have different syntactic roles and/or *ezafe* acceptance criteria. While conducting the error analysis, we discovered considerable amounts of errors in Bijankhan corpus, which motivated us to correct the *ezafe* labels of a part of the test corpus and measure the performance again. We, therefore, asked two annotators to re-annotate *ezafe* labels of the first 500 sentences of the test corpus in parallel, and a third annotator's opinion where there is a disagreement. The results of the best model, XLMRoBERTa, on the first 500 sentences of the test corpus before and after the *ezafe* label correction can be seen in Table 5. These 500 sentences contain 14,934 words, 3,373 of them with *ezafe*, based on Bijankhan labels.

Test Corpus	**Precision**	**Recall**	**F_1-score**
Bijankhan	0.9691	0.9851	0.9770
Corrected	0.9838	0.9897	0.9867

Table 5: XLMRoBERTa's precision, recall, and F_1-score on the first 500 sentences of the test set, before and after *ezafe* label correction.

Correcting *ezafe* labels resulted in 0.97% increase in F_1-score on the abovementioned part of the test corpus. The same correction for all of the test corpus might result in a near 99% F_1-score for XLMRoBERTa model. Transformer-based models perform remarkably even where there is a typo crucial to *ezafe* recognition, i.e., when the intrusive consonant '*y*' is missed between an ending vowel and a (not-written) *ezafe*, for instance, *diskhā-**y**[e]* "disks" and *be'ezā-**y**[e]* "for".

6 POS Tagging

For the task of POS tagging, we experimented with CRF_1, CRF_2, BLSTM+CNN, BERT, and XLM-RoBERTa models in the single-task settings, multi-task settings with *ezafe* as the auxiliary task (except for CRFs), and also in a single-task setting with *ezafe* information in the input. For the last one, we added the *ezafe* output of XLMRoBERTa in Section 5 to the input text. In this section, we first explain the role of *ezafe* information in POS tagging, then we discuss the results of the POS tagging task, and then we analyze it.

6.1 The Role of *Ezafe*

Ezafe is a linker between words in nonverbal phrases. It is hence not used between phrases, which can be an indicator of phrase boundaries (Tabibzadeh, 2014). Compare Examples 5a and 5b, for instance. This means that *ezafe* information will help the model, and also humans, to better detect the phrase boundaries, which can be helpful in recognizing syntactic roles (Nourian et al., 2015).

(5) a. [*pesar*] [*xošhāl*] [*'āmad*]
boy happy came
N ADV V
"The boy came happily"

b. [*pesar-e* *xošhāl*] [*'āmad*]
boy-*ez.* happy came
N ADJ V
"The happy boy came"

Knowing *ezafe* also helps the model determine the POS of some homographs. Some examples are as follows. The information below is resulted from studying homographs based on their POSs in Bijankhan corpus.

- The '*i*' suffix in Persian can be derivational or inflectional. When derivational, it is either a nominalizer or an adjectivizer and the derived form will accept *ezafe*. When inflectional, it is an indefinite marker and the inflected form will not accept *ezafe*. Some examples are *kamyābi* "scarcity, rarity", *yeksāni* "sameness", *šegeft'angizi* "wonderfulness", *bimāri* "illness", *'āšpazi* "cooking".

- Adverbized adjectives that are homonyms in both roles, accept *ezafe* only in their adjec-

	Model	Validation				Test				Approx.
		Prec.	Recall	F_1	Acc.	Prec.	Recall	F_1	Acc.	T.T.
Single	**CRF_1 (baseline)**	0.9688	0.9380	0.9521	0.9832	0.9680	0.9373	0.9511	0.9831	0.8 h
	CRF_2	0.9679	0.9530	0.9602	0.9854	0.9684	0.9514	0.9595	0.9854	0.9 h
	BLSTM+CNN	0.9680	0.9573	0.9626	0.9873	0.9677	0.9570	0.9623	0.9869	1.3 h
	BERT	0.9703	**0.9719**	0.9710	0.9899	0.9687	**0.9716**	0.9701	*0.9895*	1.4 h
	XLMRoBERTa	0.9700	0.9718	*0.9708*	**0.9900**	0.9706	0.9714	**0.9709**	**0.9901**	0.9 h
Input	**CRF_2**	0.9697	0.9563	0.9628	0.9859	0.9708	0.9555	0.9629	0.9859	1 h
	BLSTM+CNN	0.9724	0.9597	0.9660	0.9878	0.9731	0.9587	0.9658	0.9877	1.4 h
	BERT	0.9731	*0.9691*	**0.9711**	*0.9897*	0.9710	*0.9690*	*0.9700*	0.9897	1.5 h
	XLMRoBERTa	*0.9730*	0.9689	0.9709	0.9896	*0.9714*	0.9692	0.9703	*0.9895*	1 h
Multi	**BLSTM+CNN**	0.9727	0.9569	0.9647	0.9875	0.9724	0.9565	0.9643	0.9872	1.4 h
	BERT	**0.9735**	0.9665	0.9699	0.9896	**0.9728**	0.9650	0.9688	0.9888	1.5 h
	XLMRoBERTa	*0.9730*	0.9656	0.9692	0.9887	0.9725	0.9648	0.9686	0.9884	1 h

Table 6: POS tagging results (precision, recall, F_1-score, and accuracy) on the validation and test sets using the single- and multi-task and *ezafe* in the input settings. In each column, the best result(s) is/are in bold, the second best underlined, and the third best italicized. The last column shows the approximate training time in hours.

tive role. For example *samimāne* “friendly, cordial” and *ma’refatšenāsāne* “epistemological”.

- Determiners that have a pronoun form accept *ezafe* in the former, but not in the latter role. For example *’aqlab* “mostly”, *’aksar* “most of”, *hame* “all”, *’omum* “general, most of”.

- *Ezafe* information might also help the model better recognize POSs that never accept *ezafe*, such as verbs (V) and identifiers (IDEN).

6.2 Results

Table 6 shows the results of POS tagging on validation and test sets using single- and multi-task and *ezafe* in the input settings. With the single-task settings, XLMRoBERTa and BERT outperform the other models and have almost equal performances. When *ezafe* information is fed to the input, the precision of all the models increases while the recall shows a more complex behavior. For CRF_2 and BLSTM+CNN, we see a slight increase, and for the transformer-based models, we see a decrease of 0.3 to 0.4%. The F_1-score of CRF_2 model increases by 0.34% and BLSTM+CNN model by 0.27%. For BERT, it stays almost the same, and for XLMRoBERTa, it sees a decrease of 0.06%. Table 7 shows the change in F_1-scores of each POS when *ezafe* is fed with the input. As for the multi-task settings, the precision goes up, and the recall and the F_1-score come down for transformer-based and BLSTM-CNN models. Figure 3 shows POS tagging F_1-scores for single-task, in the inputs, and multi-task settings, respectively, from top to bottom, on the test set.

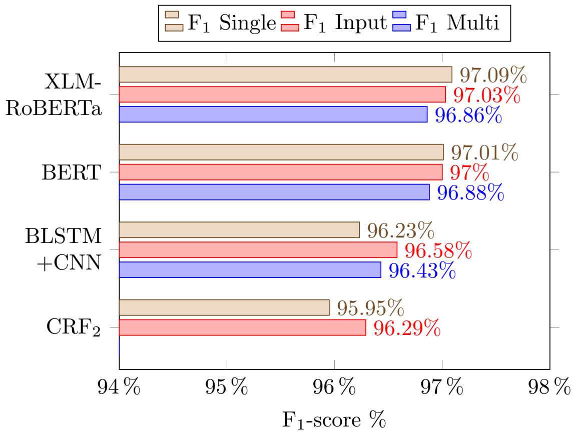

Figure 3: POS tagging F_1-scores for single-task, input, and multi-task settings, respectively from top to bottom, on the test set.

Table 8 shows POS tagging F_1-scores per POS on the test set for the single-task and *ezafe* in the input settings for CRF_2 and BLSTM+CNN models and for single-task settings for XLMRoBERTa model. An increase can be seen in the F_1-score when *ezafe* information is provided to the model. As there is no increase in XLMRoBERTa’s results when *ezafe* information is provided, the results for this setting are not shown for this model.

POS	CRF_2	B.+C.	POS	CRF_2	B.+C.
IDEN	2.80%	2.69%	**ADJ**	0.05%	0.07%
FW	0.79%	0.83%	**P**	0.03%	0.06%
ADV	0.64%	0.69%	**N**	0.03%	0.03%
DET	0.13%	0.16%	**NUM**	0.02%	0.02%
V	0.06%	0.15%	**CON**	0.01%	0.00%
PRO	0.06%	0.08%	**DELM**	0.00%	0.00%
MISC	0.06%	0.08%	**PSTP**	0.00%	-0.01%

Table 7: The change in POS tagging F_1-scores for CRF_2 and BLSTM+CNN models when *ezafe* information is fed with the input.

POS	CRF_2		BLSTM+CNN		X.R.
	Single	Input	Single	Input	Single
DELM	0.9999	0.9999	1.0000	1.0000	1.0000
PSTP	0.9995	0.9995	0.9996	0.9995	0.9998
NUM	0.9964	0.9966	0.9974	0.9982	0.9969
CON	0.9949	0.9950	0.9964	0.9964	0.9968
P	0.9944	0.9947	0.9959	0.9961	0.9966
V	0.9943	0.9949	0.9958	0.9964	0.9965
N	0.9870	0.9873	0.9893	0.9896	0.9904
PRO	0.9711	0.9717	0.9788	0.9795	0.9835
DET	0.9661	0.9674	0.9705	0.9713	0.9784
ADJ	0.9519	0.9524	0.9539	0.9555	0.9635
ADV	0.9300	0.9364	0.9414	0.9483	0.9534
MISC	0.9117	0.9123	0.9127	0.9142	0.9375
FW	0.9046	0.9125	0.9036	0.9119	0.9337
IDEN	0.8318	0.8598	0.8375	0.8644	0.8656

Table 8: POS tagging F_1-scores per POS on the test set for CRF_2 and BLSTM+CNN (single-task and *ezafe* in the input) and for XLMRoBERTa (single-task).

6.3 Analysis

As discussed in Subsection 6.1, we anticipated to see an increase in several POSs, including N, ADJ, ADV, DET, V, and IDEN. According to Table 8, the highest increase belongs to IDEN, FW, ADV with an average increase of ∼2.75%, ∼0.81%, and ∼0.67%, respectively. The increase for V is 0.06% and for N, 0.03% for both models, and for DET, 0.13% and 0.08%, and for ADJ, 0.05% and 0.16% for CRF_2 and BLSTM+CNN, respectively.

As for the transformer-based models results, they do not seem to benefit from the *ezafe* information either in the input or as an auxiliary task. As the work on syntactic probing shows, attention heads in transformer-based models, specifically BERT, capture some dependency relation types (Htut et al., 2019). As *ezafe* is a more limited form of dependency (Nassajian et al., 2019), its information could be captured by the attention heads in such models. On the other hand, contextualized embeddings also seem to capture some syntactic relations (Tenney et al., 2019; Hewitt and Manning, 2019), which is another reason for such models' high performance in capturing *ezafe* information.

All in all, it seems that transformer-based models already have captured the *ezafe* information owing to their architecture (attention heads), pretraining, contextual embeddings, and finally, being trained on the POS tagging task (which is related to the task of *ezafe* recognition, and that is why their performance does not enhance when such information is provided.

7 Conclusion and Future Work

In this paper, we experimented with different models in the tasks of *ezafe* recognition and POS tagging and showed that transformer-based models outperform the other models by a wide margin. We also provided *ezafe* information to the POS tagging models and showed that while CRF and RNN-based models benefit from this information, transformer-based models do not. We suggest that this behavior is most probably due to (1) contextual representation, (2) pretrained weights, which means a limited knowledge of syntactic relations between words, (3) the attention heads in these models, and (4) being trained on the POS task, which is related to *ezafe* recognition. An interesting direction for future work would be to investigate the role of *ezafe* in transformer-based models in the tasks that such information would be helpful, such as dependency and shallow parsing.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Jan W Amtrup, Hamid Mansouri Rad, Karine Megerdoomian, and Rémi Zajac. 2000. Persian-english machine translation: An overview of the shiraz project. *Memoranda in Computer and Cognitive Science MCCS-00-319, NMSU, CRL.*

Habibollah Asghari, Jalal Maleki, and Heshaam Faili. 2014. 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.

Ali Azimizadeh, Mohammad Mehdi Arab, and Saeid Rahati Quchani. 2008. Persian part of speech tagger based on hidden markov model. *9th JADT.*

Sara Bahaadini, Hossein Sameti, and Soheil Khorram. 2011. Implementation and evaluation of statistical parametric speech synthesis methods for the persian language. In *2011 IEEE International Workshop on*

Machine Learning for Signal Processing, pages 1–6. IEEE.

Mahmood Bijankhan. 2004. The persian text corpus. In *In 1st Workshop on Persian Language and Computer. Tehran*.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Masood Ghayoomi and Saeedeh Momtazi. 2009. Challenges in developing persian corpora from online resources. In *2009 International Conference on Asian Language Processing*, pages 108–113. IEEE.

Jila Ghomeshi. 1997. Non-projecting nouns and the ezafe: construction in persian. *Natural Language & Linguistic Theory*, 15(4):729–788.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*. Morgan & Claypool Publishers.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Anders Holmberg and David Odden. 2005. The noun phrase in hawramani, presented at the. In *First International Conference on Aspects of Iranian Linguistics*.

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

Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

S Isapour, M Homayounpour, and M Bijabkhan. 2008. 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*.

Taylor Jones. 2018. An argument for ezafe constructions and construct state in zulu. *Proceedings of the Linguistic Society of America*, 3(1):58–1.

Arsalan Kahnemuyipour. 2006. Persian ezafe construction: case, agreement or something else. In *Proceedings of the 2nd Workshop on the Persian Language and Computer. Tehran: University of Tehran*.

Arsalan Kahnemuyipour. 2014. Revisiting the persian ezafe construction: A roll-up movement analysis. *Lingua*, 150:1–24.

Arsalan Kahnemuyipour. 2016. The ezafe construction: Persian and beyond. In *Conference on Central Asian Languages and Linguistics*, page 3.

Simin Karimi and Michael Brame. 2012. A generalization concerning the ezafe construction in persian. *Linguistic Analysis*, 38(1):111–144.

Yadgar Karimi. 2007. Kurdish ezafe construction: Implications for dp structure. *Lingua*, 117(12):2159–2177.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Mikhail Korobov. 2015. sklearn-crfsuite.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Richard Larson and Hiroko Yamakido. 2008. Ezafe and the deep position of nominal modifiers. *Adjectives and adverbs: Syntax, semantics, and discourse*, pages 43–70.

Richard K Larson and Hiroko Yamakido. 2006. Zazaki "double ezafe" as double case-marking. In *annual meeting of the Linguistic Society of America, Albuquerque, NM*.

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

Mahdi Mohseni and Behrouz Minaei-Bidgoli. 2010. A persian part-of-speech tagger based on morphological analysis. In *LREC*.

Salar Mohtaj, Behnam Roshanfekr, Atefeh Zafarian, and Habibollah Asghari. 2018. Parsivar: A language processing toolkit for persian. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Stefan Müller and Masood Ghayoomi. 2010. Pergram: A trale implementation of an hpsg fragment of persian. In *Proceedings of the International Multiconference on Computer Science and Information Technology*, pages 461–467. IEEE.

Minoo Nassajian, Razieh Shojaei, and Mohammad Bahrani. 2019. The corpus-based study of ezafe construction in persian.

Samira Noferesti and Mehrnoush Shamsfard. 2014. A hybrid algorithm for recognizing the position of ezafe constructions in persian texts. *IJIMAI*, 2(6):17–25.

Peyman Nojoumian. 2011. *Towards the Development of an Automatic Diacritizer for the Persian Orthography based on the Xerox Finite State Transducer*. University of Ottawa (Canada).

Alireza Nourian, Mohammad Sadegh Rasooli, Mohsen Imany, and Heshaam Faili. 2015. On the importance of ezafe construction in persian parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 877–882.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Mohammad Javad Rezai and Tayebeh Mosavi Miangah. 2017. Farsitag: A part-of-speech tagging system for persian. *Digital Scholarship in the Humanities*, 32(3):632–642.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Benoît Sagot and Géraldine Walther. 2010. A morphological lexicon for the persian language.

Vida Samiian. 1994. The ezafe construction: Some implications for the theory of x-bar syntax. *Persian Studies in North America*, pages 17–41.

Pollet Samvelian. 2006. When morphology does better than syntax: The ezafe construction in persian. *Ms., Université de Paris*, 3.

Pollet Samvelian. 2007. The ezafe as a head-marking inflectional affix: Evidence from persian and kurmanji kurdish.

Claude E Shannon. 1948. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423.

Mansour Sheikhan, M Tebyani, and M Lotfizad. 1997. Continuous speech recognition and syntactic processing in iranian farsi language. *International Journal of Speech Technology*, 1(2):135–141.

Ian F Spellerberg and Peter J Fedor. 2003. A tribute to claude shannon (1916–2001) and a plea for more rigorous use of species richness, species diversity and the 'shannon–wiener'index. *Global ecology and biogeography*, 12(3):177–179.

Omid Tabibzadeh. 2014. Persian grammar: a theory of autonomous phrases based on dependency grammar. In *Nasre Markaz. Tehran*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Maziar Toosarvandani and Coppe van Urk. 2014. The syntax of nominal concord: What ezafe in zazaki shows us. In *Proceedings of NELS*, volume 43, pages 221–234.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

M. S. Zahedi, M. H. Bokaei, F. Shoeleh, M. M. Yadollahi, E. Doostmohammadi, and M. Farhoodi. 2018. Persian word embedding evaluation benchmarks. In *Electrical Engineering (ICEE), Iranian Conference on*, pages 1583–1588.