Journal of Information Systems and Telecommunication
Vol.12, No.1, January-March 2024, 72-89
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

# Persian Ezafe Recognition Using Neural Approaches

Habibollah Asghari[1]*, Heshaam Faili[2]

[1.] ICT Research Institute, ACECR, Tehran, Iran,
[2.] Department of ECE, School of Engineering, University of Tehran, Tehran, Iran.

Received: 18 Apr 2023/ Revised: 24 Feb 2024/ Accepted: 13 Mar 2024

## Abstract

Persian Ezafe Recognition aims to automatically identify the occurrences of Ezafe (short vowel /e/) which should be pronounced but usually is not orthographically represented. This task is similar to the task of diacritization and vowel restoration in Arabic. Ezafe recognition can be used in spelling disambiguation in Text to Speech Systems (TTS) and various other language processing tasks such as syntactic parsing and semantic role labeling.

In this paper, we propose two neural approaches for the automatic recognition of Ezafe markers in Persian texts. We have tackled the Ezafe recognition task by using a Neural Sequence Labeling method and a Neural Machine Translation (NMT) approach as well. Some syntactic features are proposed to be exploited in the neural models. We have used various combinations of lexical features such as word forms, Part of Speech Tags, and ending letter of the words to be applied to the models. These features were statistically derived using a large annotated Persian text corpus and were optimized by a forward selection method.

In order to evaluate the performance of our approaches, we examined nine baseline models including state-of-the-art approaches for recognition of Ezafe markers in Persian text. Our experiments on Persian Ezafe recognition based on neural approaches employing some optimized features into the models show that they can drastically improve the results of the baselines. They can also achieve better results than the Conditional Random Field method as the best-performing baseline. On the other hand, although the results of the NMT approach show a better performance compared to other baseline approaches, it cannot achieve better performance than the Neural Sequence Labeling method. The best achieved F1-measure based on neural sequence labeling is 96.29%

**Keywords:** Persian Ezafe Recognition; Vowel Restoration; Diacritization; Neural Sequence Labeling.

## 1- Introduction

In the Persian language, Ezafe construction is an unstressed short vowel /-e/ (or /-ye/ after vowels) which is used to link two words in some contexts. The function of Ezafe in Persian is to link elements of an NP or PP in (a) possessive constructions, (b) modification of the noun, (c) connecting some of the preposition types to the following NP elements, and (d) Proper Names linking first name to last name [1].

Although Ezafe is an important part of Persian phonology and morphology, it does not have a specific orthographical representation and it is not usually written. However, it should be pronounced as a short vowel /e/. Ezafe appears between two words indicating some relationships between the words to show the occurrence of a genitive case. Sometimes, its presence is explicitly marked by the diacritic "Kasre" in order to facilitate the correct pronunciation. Common uses of the Persian Ezafe are pronominal possession, possessive suffixes, adjective-nouns, and title-family names. We will discuss the various cases of Ezafe connecting two words in more detail in Section 3.

In most cases, despite the lack of orthographical representation, the location of Ezafe can be identified by human readers. However, very often, automatic recognition of Ezafe markers is a challenging problem. One of the challenges in Persian language processing is to determine how this important unwritten short vowel should be recognized.

The most important application of Ezafe identification is in the context of text to speech (TTS) systems as a text to phoneme tool [2]. Other applications include Ezafe recognition for identifying the dependency of a word in a

✉ **Habibollah Asghari**
habib.asghari@ictrc.ac.ir

Noun Phrase [3], or back-and-forth transliteration from the Perso-Arabic writing system and Latin-based scripts [4]. It has also been shown that adding information regarding Ezafe markers can greatly improve dependency parsing and shallow parsing as well [5].

Some tagging algorithms in computational linguistics can be applied to accomplish the task of Ezafe recognition. Various rule-based and statistical approaches for recognition of Ezafe markers have been investigated including HMM POS tagger [3], Maximum Entropy (MaxEnt) POS tagger, Conditional Random Fields (CRF) tagger [6], phrase-based Statistical Machine Translation [6], and Genetic Algorithms [7]. NLP techniques such as Probabilistic Context-Free Grammars (PCFGs) are also good tools for finding NPs and searching Ezafe construction inside the constituent [8].

In this research, we would like to investigate various approaches to automatically recognize the location of Ezafe construction in the Persian language. Our main focus is on neural approaches, including neural sequence labeling and neural machine translation (NMT) as well. In modeling the Ezafe recognition task as a translation problem, the input to the system is a Persian text without Ezafe markers, and the output of the algorithm is the same text marked with Ezafe tags. To evaluate the performance of our proposed methods, we conducted various baseline experiments and previous approaches and compared them with our approach. To the best of our knowledge, no work has been done so far to investigate neural approaches to the problem of Persian Ezafe recognition.

In a short view, the contributions of this paper are as follows:

- The use of a Neural sequence labeling approach that is state-of-the-art in Ezafe recognition.
- Using a Neural Machine Translation (NMT) approach.
- Incorporating versatile syntactic and lexical features of the Persian language in the Ezafe recognition task and embedding them in our models as a whole.
- The use of a large corpus for training the system, so the results are considerably reliable.

In addition, we investigated a Factored–based SMT model (FB-SMT) as an extension to the statistical machine translation model that was previously developed by [6] for recognizing Ezafe in Persian. To this end, we applied various features to enhance the functionality of SMT in Ezafe recognition.

The paper is organized as follows. In the next section, a clear definition of the problem is presented and the characteristics of the Persian language are introduced. In Section 3, we will give a precise definition of Ezafe and its role in the Persian language. Section 4 provides an overview of previous works in Ezafe recognition. Our approach will be described in Section 5, in which we will explain the proposed neural models and also the features that are incorporated into them. Experiments and results will be provided in Section 6 including experimental setup, evaluation measures, the baselines, and implementation of our proposed method. Discussion and analysis of the results will be explained in Section 7. Conclusion and recommendations for future works will be discussed in the final section.

## 2- An Overview of Persian Language

Persian is a rich morphology language that belongs to Arabic script-based languages. This category of languages includes Kurdish, Urdu, Arabic, Pashtu, and Persian [9]. They all have common scripting and somehow similar writing systems. This language family has some common properties and features such as the absence of capitalization, right-to-left direction, encoding issues in the computer environment, lack of clear word boundaries in multi-token words, and a high degree of ambiguity due to non-representation of short vowels in the writing system [9]. Note that Ezafe recognition and homograph disambiguation problems mostly deal with the last two mentioned features.

It should be noted that despite the mentioned commonalities, these languages do not belong to a single language family. Although Persian and Arabic have almost the same scripts and share some common characteristics, Persian belongs to the Indo-European language family, while Arabic is a Semitic language, belongs to the Afro-Asiatic family of languages, completely different in lexical and syntactic features [10]. For example, Urdu and Arabic have grammatical gender determiners, while Persian does not have a gender marker. There are also some word order differences, for example, while Arabic has predominantly SVO order, Persian and Urdu languages follow SOV order [11, 12]. Persian is the official language of three countries including Iran, Tajikistan, and Afghanistan.

## 3- Ezafe Definition

In the Persian language, the elements within a noun phrase are linked by an enclitic particle called Ezafe. This morpheme is usually an unwritten vowel, but it could also have an orthographic realization. In most cases, this relation can be translated as a genitive structure. An example of this construction is as follows:

- */ketâb -e mæn/*
- */book-EZ I/*
- *my book*

Note that in the ordinary Persian writing system, the short vowel /-e/ is not written, and this causes ambiguities in pronunciation. It should be mentioned that the Ezafe is not

typically written when it follows a consonant or glide (i.e., 'ye'), but it is overtly written when it follows a vowel (i.e., /a/, /u/, /i/). It is also sometimes overtly written following the final (silent) 'he' in current writing, where the Ezafe can be written as a separate 'ye'. Ezafe is a property of the Arabic script used in Iran and Afghanistan, while it is written overtly in Tajiki Persian which employs the Cyrillic script.

While reading text, native speakers can generally vocalize each word based on their familiarity with the lexicon and the context of the text. However, it is hard to automatically recognize Ezafe in ordinary text because of considerable ambiguity. In recognizing Ezafe, the computer program should take into account morphological, syntactic, semantic, and discourse views [13].

There is also Ezafe construction in other languages; for example Ezafe construction in Urdu, which borrowed the construction from Persian [14].

Historically, the origin of enclitic Ezafe was in a demonstrative-relative morpheme in old Iran [15]. In Persian, it can be related to a demonstrative /hya/, which links the head noun to adjectival modifiers in possessor NP in Old Persian [16]. In the evolution of the Persian language, /hya/ changed to /–i/ in Middle Persian and progressively lost its demonstrative value to end up as a simple linker [16]. Contrary to Persian, Kurdish and Zazaki still have a so-called "Demonstrative Ezafe", different from the affixal Ezafe, which functions as a demonstrative pronoun heading nominal phrases.

Ezafe can appear in noun phrases, adjective phrases, and some prepositional phrases linking head and modifiers. It should be stated that Ezafe is limited to specific POS tags such as N, ADJ, P, NUM, DET, ADV, and PRO respectively. So for example, it cannot appear on 'yek' in the above two examples, and it would rarely appear on a pronoun, which limits the permutations [17].

## 3-1- Ezafe iteration

Ezafe can be iterated within NPs, occurring as many times as there are modifiers [16]. In the following example, fourteen words are related to each other by iterated Ezafe markers:

- */Lozum-e tæqyir-e zæmân-e bærgozâri-ye dour-e moqqædæmâti-ye mosâbeqât-e futbâl-e jâm-e jæhâni-ye sâl-e 2010-e âfriqâ-ye jonubi/*
- /need-EZ change-EZ time-EZ hold-EZ round-EZ preliminary-EZ competition-EZ soccer-EZ cup-EZ world-EZ year-EZ 2010-EZ Africa-EZ south/
- /The need to change the time of holding the preliminary round of South Africa's year 2010 soccer World Cup competition /

As can be shown in the above sentence, the chain of Ezafe markers can iterate within a phrase linking several elements together. As a result, there is no limitation in the number of words in a connected chain of Ezafe markers.

## 3-2- Ezafe Domain definition

We refer to all elements that are consecutively linked by the Ezafe marker as "Ezafe domain". Ezafe domain is a specific phrase domain comprised of all the words that relate to each other using Ezafe. Determination of the Ezafe domain is equal to determining the Ezafe location between words [18, 3]. So the task of Ezafe identification is to correctly and accurately define the Ezafe domain boundaries.

Assume that /w1 w2 w3 ... wn/ is a Persian sentence. In this sentence, the sequence wi ... wi+j is an Ezafe domain if all of the words in the sequence have Ezafe as /w1 w2 … wi-e wi+1 -e wi+2 -e ... wi+j-1 -e …wn /

## 3-3- Problems with Ezafe Marking

There are some problems with automatically recognizing Ezafe markers in Persian text. It is because in recognizing Ezafe, we should consider morphological, syntactic, semantic, and discourse views [13]. The following examples show the importance of each of the mentioned views:

The example below needs morphological analysis to find the location of Ezafe marker (represented as -ye in the example):

- */hæme-ye osærâ-ye jæng âzâd ʃodænd/*
- */all-EZ captives-EZ war were freed/*
- *All of the captives of the war were freed*

As mentioned before, when the last letter of the word is a vowel, then the Ezafe marker changes to /-ye/ instead of /e/.

Sometimes we need syntactic analysis to locate the Ezafe marker in the sentence [56]. In this example, we need to analyze the status of the verb in the sentence to find the exact position of the Ezafe marker:

- */yek mærd-e dɒ:neʃmænd râ didæm/* (I saw a scientist man)
- */yek mærd dɒ:neʃmænd râ did/* (A man saw a scientist)

In the example above, both sentences are written in the exact same way, but should be pronounced differently. In the first sentence, a pronoun drop has occurred; this can be derived by analyzing the verb 'didam'. In the second sentence, two meanings can be deducted based on the location of the Ezafe marker. If an Ezafe marker exists on 'mærd', then a pronoun drop for the subject 'he' has occurred, and 'mærd' is an object. On the other hand, if no Ezafe marker exists, then 'mærd' is a subject.

The example below needs semantic analysis. Notice that the pronoun has been dropped from the second sentence and so, this creates an ambiguity in finding the subject of the sentence which results in difficulty in correctly locating the Ezafe marker:

- */æhmæd kelid-e otâq râ âværd/*
- */Ahmad the key-EZ room brought/*
- */Ahmad brought the key to the room/*
- */kelid-e otâq râ âværd/*
- */the key-EZ room brought/*
- */he/she brought the key to the room/*

In the above example, 'ahmad', the subject of the sentence has been omitted. The object 'kelid' can be analyzed mistakenly as the subject if it's not linked to the next element and the pronoun drop is not caught.

In the following example, in order to find the location of Ezafe marker, we need a discourse analysis:

- */Dâneʃâmuzân xâneh-ye xod râ peidâ kærdænd/*
- */Students home-EZ their found/*
- */The students found their home/*

However, to find the syntactic structure of the sentence, a discourse analysis is required. First of all, we need to recognize the role of the first word. In other words, we should discover whether it is the object or the subject of the sentence. By analyzing just the current sentence, we cannot find the correct position of Ezafe tags, so we should know what happened in the previous sentences. This example also shows the ambiguity caused by the combination of the Ezafe not being written and the SOV order in the Persian clause, since the object directly follows the subject and the boundary between the two is difficult to identify. Note that the origin of the above-mentioned problems in Ezafe recognition is mainly because of pronoun drop in the Persian language.

### 3-4- Challenges of Ezafe marking in computational linguistics

There are some challenges that we encounter in Persian language processing [59]. One of the problems in Persian language processing is related to long-distance dependencies which increase the difficulty of correctly identifying the Ezafe marker. This phenomenon complicates Ezafe recognition for humans as well; one would need to read the entire sentence before recognizing the place of Ezafe [57]. The example below shows this case (Note that /e/ in the examples stands for Ezafe marker):

- */?u xâne-ye særshâr æz æfsus væ ænduh-e xod râ tærk kærd/*
- */he home-EZ full of regret and sorrow-EZ him left/*
- *He left his house which was full of sorrow and regret.*

In the above example, note that the genitive case /xâne/ and /xod / are not next to each other, and so recognizing the presence of Ezafe in /xâne/ can be achieved by determining this long-distance dependency.

Another problem is to determine boundaries in multi-token words. In some cases, when the parts of a multi-token word are separated by a space delimiter, it is hard to recognize the Ezafe marker in the sentence.

- */?ânhâ bâ naxost vazir molâghât kardand/*
- */They with prime minister met/*
- *They met prime minister.*

In the above example, /naxost vazir/ is a multi-token word, and so /naxost/ do not need Ezafe marker.

The third challenge arises by pronoun drop due to the morphology of the Persian language. Persian is a null-subject or pro-drop language. So personal pronouns such as 'I', 'he', and 'she' are optional and can be omitted from the sentence. As can be seen in the following example, the subject can be removed from the sentence, so make it difficult to correctly recognize Ezafe in the sentence:

- */mæn âb-e khonak râ nu:ʃidam/=/âb-e khonak râ nu:ʃidam /*
- */I water-EZ cold drank/ = /water-EZ cold drank /*
- *I drank the cold water*

In the above example, a pronoun resolution is required in order to correctly find the location of the Ezafe marker.

Another challenging issue is the homograph ambiguity as a result of dropping short vowels in writing. This problem is also the origin of the main challenges we encounter in Ezafe recognition. So, Ezafe recognition can be expressed as a kind of homograph disambiguation task. The difference here is that homograph disambiguation generally deals with all of the diacritics that can be attached to the letters inside a word, but in Ezafe recognition, we are only concerned with the ending letter of the word.

Finally, another main problem in Ezafe recognition is to detect word/phase boundaries especially when we encounter multi-token words. In Persian language, affixes and words having multi tokens can be written in three kinds of writing formats; completely separated by a space delimiter, separated by a Zero Width Non-Joiner (ZWNJ) letter, or can be attached to its main word. In the first case, the computer determines them as two separate words, while in the latter two cases, the borders of words can be correctly recognized. Most of the time, Persian writers do not obey the Persian Academy rules and the writer is free to choose one of them at will. The problem of determining word boundaries makes it difficult to recognize Ezafe markers.

## 4- Related Work

In this section, we will explain the previous researches in recognizing Ezafe in the Persian language. The problem of determining short vowels in other languages such as Arabic and French is also discussed.

### 4-1- Ezafe Recognition in Persian

There have been some efforts to recognize Ezafe in the Persian language. As a first attempt to recognize Ezafe in Persian text, [18] used POS tags and also semantic labels (such as place, time, ordinal numbers ...) to obtain a statistical view of Ezafe markers. The most frequent combinations were extracted based on a 10 million-word corpus. In research accomplished by [8], the researchers focused on noun phrases. In NPs, Ezafe can relate between the head and its modifiers. Hence, by parsing the sentences and finding phrase borders, the location of the Ezafe marker in the sentence could be found. The sentences were analyzed using a Probabilistic Context Free Grammar (PCFG) to derive phrase borders. Then based on the extracted parse tree, the head and modifiers in each phrase could be determined. In the last phase, a rule-based approach was also applied to increase the accuracy of Ezafe marker labeling. There were also other attempts to effectively recognize Ezafe marker in Persian text, such as [19] based on fuzzy sets. Also, researchers in [3] developed a system based on the Hidden Markov Model to correctly identify Ezafe markers. In [20] they approached the problem using syntactic analysis. There are also some implementations using neural networks [21]. Another research for recognizing the position of Ezafe construction in Persian text has used a combined framework based on rule-based models and genetic algorithms [7]. Genetic algorithms provide a search strategy to learn general Ezafe patterns, while the rule-based model handles special cases and exceptions to general patterns. The results of this study show that the proposed algorithm outperformed classical HMM-based methods. As a last related work, researchers in [6] used three approaches named Maximum Entropy (MaxEnt) POS tagger, a Conditional Random Fields (CRF) tagger, and a phrase-based Statistical Machine Translation (PB-SMT) method. The latter approach is closely related to our approach. The difference is that we have used the FB-SMT model instead of a simple phrase-based SMT, incorporating many well-defined selected features into the model.

As a result, the Ezafe tagging problem can be classified into three categories, each of them can use algorithms at the character and/or word level. In the following subsections, we will explain them in more detail.

#### *Ezafe recognition using Rule-based tagging*

The most straightforward way to tag words with an Ezafe marker is to use some lexical or grammatical rules so as to find potential words having an Ezafe marker. This method needs human intervention by handcrafted rules. As an example, we can define some linguistic rules such as follows:

• IF the current word is NOUN and the next word is ADJ THEN (Tag the current word with an Ezafe marker)

• IF the current word has the ending letter /ی/ and the next word is ADJ THEN (Tag the current word with Ezafe marker)

Rule extraction can also be done by examining the confusion matrix. It can greatly help in evaluating false positive (FP) and false negative (FN) cases, and then derive some rules for correcting the misclassified cases of statistical methods [6].

Some of the above-mentioned related works have used a rule-based approach as a post-processing step to increase the accuracy of Ezafe recognition. In [8] the researchers used a rule-based method in the last phase of their work. Moreover, [6] applied high precision and low recall rule sets to decrease FP and FN and increase the total accuracy. They have used five Persian-specific features. Another research in [7] used a combined framework based on rule-based models and genetic algorithms.

#### *Ezafe recognition as a sequence tagging problem*

Part of Speech (POS) tagging is an effective way for automatically assigning grammatical tags to words of the sentence. There are powerful statistical POS tagger algorithms and methods. In the case of Ezafe recognition, instead of simple POS tags, an extended POS tag set is used comprised of Part of Speech tags plus Ezafe marker, which is called POSE tag that can be constructed by adding Ezafe markers to original first-level POS tags.

Among previous works, researchers in [3] has used a sequence tagging approach based on Hidden Markov Model for recognizing POSE tags. Another attempt based on sequence tagging was also done by [6] using the Conditional Random Fields (CRF) POSE tagger and also the Maximum Entropy (MaxEnt) POSE tagger. Moreover, there are some related works in word segmentation based on sequence tagging that can also be used for the task of Ezafe recognition such as the works accomplished by [22], [23], and [24] as well.

#### *Ezafe recognition as a translation problem*

Ezafe recognition problem can be considered as a translation problem; the original training text without Ezafe marker can be used as a text in the source language and the tagged text can be used as a text in the target language. In research accomplished by [6], they have used a phrase-based SMT model. Our work in this paper is also based on machine translation with a different approach.

#### *Ezafe recognition using transformers*

In a research by Doostmohammadi et al., they have exploited Transformer-based, BERT, and XLMRoBERTa

methods, and achieved the best results, with respect to the previous works [58]. In another research accomplished by Ansari et al., they tackled the problem of Ezafe recognition using ParsBert transformer. They have also compared their proposed method with the XLMRoBERTa and BERT multilingual models [60].

## 4-2- Diacritization and Vowel Restoration in Arabic

The recognition of Ezafe could be compared with prediction tasks in vowelization of Arabic. Since Persian has borrowed many words from Arabic, so Ezafe recognition in some ways is similar to diacritization in Arabic. As mentioned before, there are some similarities in Arabic-script-based languages. For example, in the Arabic language there is an ambiguity resulting from the absence of short vowel representations in contemporary Arabic texts [25]. Arabic readers infer the appropriate diacritics based on linguistic knowledge and the context. However, in the case of text-to-speech or automatic translation systems, Arabic letters need to be diacritized. Otherwise, the system will not be able to know which word to select. Like Persian, vowel restoration of Arabic text is also a homograph disambiguation process. The restoration of diacritics in written Arabic is an important processing step for several natural language processing applications [26]. In Arabic, there are eight diacritics as shown in Table 1. A diacritic may be placed above or below a letter hinting how the letter should be pronounced. In other words, they are written above or below the consonants they follow. The first three diacritics represent the Arabic short vowels. In our case in Persian Ezafe recognition, the third one in the table is important.

Table 1: Diacritics in Arabic Language

| No. | Diacritic Shape | Diacritic Name | Location of diacritic | Pronunciation |
|---|---|---|---|---|
| 1. | ◌َ | *Fathah* | کَ | /Ka/ |
| 2. | ◌ُ | *Dhammah* | کُ | /Ku/ |
| 3. | ◌ِ | *Kasrah* | کِ | /Ke/ |
| 4. | ◌ً | *Tanweenfathah* | کً | /Kan/ |
| 5. | ◌ٌ | *Tanweendhammah* | کٌ | /Kun/ |
| 6. | ◌ٍ | *Tanweenkasrah* | کٍ | /Kin/ |
| 7. | ◌ّ | *Shaddah (Double consonant marker)* | کّ | /KK/ |
| 8. | ◌° | *Sukoon* | ک° | /K/ |

As an example of Arabic diacritization, [27] proposes an approach based on HMM to solve the problem of automatic generation of diacritical marks of the Arabic text. The system should be trained based on a specific topic, e. g. sports, weather, local news, international news, business, economics, religion, etc. For testing purposes, they have used a fully diacritized transcript of the Holy Quran. The recognition rate was about 95.9%.

There is also another research done based on the effect of Arabic language diacritization on Statistical Machine Translation. It is shown that this method outperforms the previous methods [28]. In another research for diacritization of Arabic text using morphological tagging, they used lexical resources [29]. They have stated that Out Of Vocabulary (OOV) words such as foreign words and names can affect the system. Diacritization using deep neural approaches has also been investigated in Arabic (Belinkov and Glass 2015, Abandah, et al., 2015) [30, 31]. They have used recurrent neural networks for the task of vowel restoration. Another research introduces an approach for Arabic diacritization utilizing Bidirectional Encoder representations from Transformers (BERT) models [61]. The performance of the model was assessed using various error metrics, including Diacritic Error Rate (DER) and Word Error Rate (WER).

A literature review of the previous works in automatic diacritics restoration in Arabic has been accomplished by Lapointe, et al, [62].

## 4-3- Other Languages

There are also some closely related problems in other languages. One of them would be the liaison in French grammar. Liaison is a grammatical circumstance in which a usually silent consonant at the end of a word is pronounced at the beginning of the word that follows it. Part of the reason that French pronunciation and aural comprehension are somehow difficult is due to liaisons [32].

In the Caspian languages Gilaki and Mazandarani, and in neighboring languages like Taleshi, nominals are near mirror inverses of Persian; a wide range of noun complements occur pre-nominally, and link to N via “reverse Ezafe” particle that can be shown by -REZ [33]. The following example shows this phenomenon:

- */surx-e Gul/*
- */red-REZ flower/*
- */red flower/*

Moreover, by comparing Chinese to this family of Iranian languages, which show rich variation in nominal structure, it can be shown that Chinese /de/ has the essential properties of a reverse Ezafe particle, as exemplified by the Caspian languages Gilaki and Mazandarani [33].

## 5- Our Approach

In this paper we deal with neural models to tackle the problem of Ezafe recognition. In the first approach, two models of neural sequence taggers are employed for the task of Ezafe recognition. In the second approach, a neural machine translation approach is used. Neural machine translation has shown its performance in machine translation problems. We compare the performance of our two approaches with that of the baselines.

Recognition of Ezafe construction heavily depends on the surrounding context. By incorporating good features, they can effectively present the context into the model, and so enhance the recognition rate of Ezafe markers in the text.

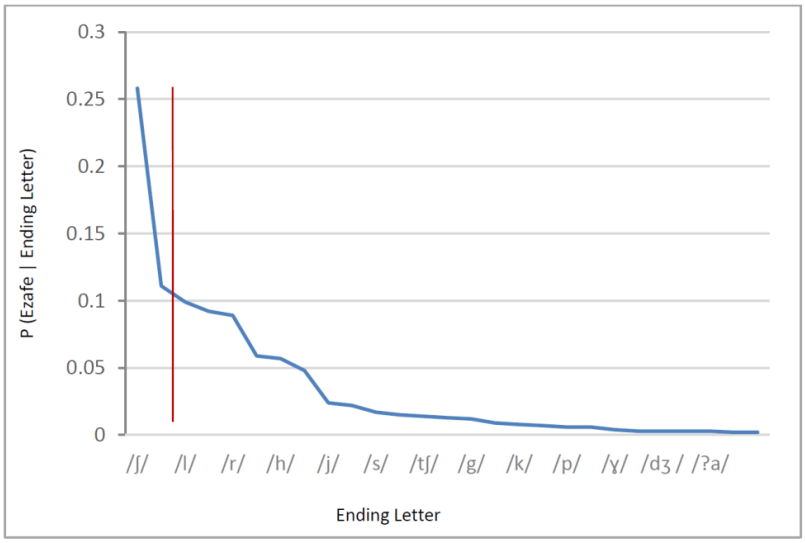


Fig. 1: Conditional probability of Ezafe appearance vs. ending letters

One of the advantages of our approach is to construct an integrated model that includes all the features as a whole, whereas, in the previous research, a two-stage task comprised of a statistical model followed by a rule-based post-processing step is employed. In our approach, all of the features (including the Persian-specific features) are included into one integrated model. As a result, the optimization of the features is more flexible in our model.

In the following subsections, we first describe the syntax-aware features that we have used in our model. Then, in the two subsequent sections, we present the two above-mentioned neural approaches.

### 5-1- Features for Ezafe Recognition

In this paper, we have applied some well-behaved syntax-aware features and examined that if they can improve the performance of Persian Ezafe recognition. Since Persian Ezafe Recognition is a Syntax-oriented task, so we selected the best combination of lexical and grammatical Persian linguistic features for incorporating them into our model.

In the training phase, some features are used into the system along with a large amount of annotated text data. It should be noted that the selection of the features is a very important task and should be done precisely. The features that we have selected for the recognition of Ezafe markers are as follows:

- POS tag of the current word: The existence of Ezafe tags greatly depends on the POS tag of the current word.
- POS tag of the next word: The POS tag of the next word is also of great importance because the Ezafe marker occurs between the current and the next word.
- P(Ezafe|word) with different values for each word: This feature is a maximum likelihood estimate by calculating all the words having Ezafe marker in the training corpus.
- P(Ezafe|POS) with different values for each POS tag: This feature is a maximum likelihood estimate by calculating all POS tags having Ezafe marker in the training corpus.
- POS tag of the previous adjacent word: Our experiments have shown that Part of Speech tags of the previous word can be used as an effective feature in Ezafe recognition.
- POS tags of the two next adjacent words: The experiments have shown that Part of Speech tags of

the second next word can be used as a good feature in Ezafe recognition.

The effect of the ending letter of the target word was also examined as a feature in the model. In our experiments, it was shown that the ending letter of the target word is of great importance in Ezafe recognition. So we tried various features that can be constructed based on ending letters. The Persian language has 32 letters, and they may all appear as the ending letters of a word. Moreover, some diacritics could appear at the end of some words and should be taken into account as ending characters. As a result, 46 ending characters (comprised of 32 ending letters plus 14 types of diacritics) exist in our corpus. To test the effectiveness of the ending letters feature, we calculated *P(Ezafe|EndingLetter)*, and then selected the most important ones as shown in Fig. 1. The graph shows the likelihood of the Ezafe marker with respect to the ending letters. As shown in the figure, some letters make a high probability on *P(Ezafe|EndingLetter)*.

By considering the breakpoint in the curve, the nine letters with the highest conditional probability were selected, which are */ʃ/, /m/, /l/, /d/, /r/, /t/, /h/, /n/, /j/* respectively. We call these highly important letters as golden ending letters.

As a result, three sets of features were constructed based on ending letters as follows:

- Ending letter of the current word (one hot representation of length 32): By assigning 32 different features for each word, this feature can take binary values for each letter.
- Ending letter of the current word (showed by one byte of binary coded format): By assigning one feature for each word, this feature can take 32 different values for each letter.
- Golden Ending letters: The high probable ending letters of the words that can take Ezafe markers can also be used as a feature. In this paper we have called them as golden letters and they are used as binary features (9 factors with binary values).

In order to find the best feature set, various combinations of these features were examined. Table 2 demonstrates the combination of features that have been investigated to be applied to our models. An approach would be to use Forward Selection procedure to obtain the best variables and then incorporate them into the models. In selecting the best set of features, we applied the Forward Selection method; taking into account what variables are eligible to be added to the set of features. This method is often used to provide an initial screening of the candidate variables when a large group of variables exists. As a result, the Ending letter (non-binary feature) and Golden Ending letters (binary features) were removed from the final feature set.

As an example, feature set 3 comprised of four features including likelihood of the word to take the Ezafe marker, the POS tag of the target word, the likelihood of the POS to take the Ezafe marker, and the ending letter (binary factors) were used as factors into the system.

## 5-2- Ezafe Marking by Neural Sequence Labeling

With the advances in deep learning, neural sequence labeling models have achieved state-of-the-art for many tasks [34, 23, and 35]. Features are extracted automatically through network structures including long short-term memory (LSTM) [36], and convolution neural network (CNN) [37] with distributed word representations. Similar to discrete models, a CRF layer is used in many state-of-the-art neural sequence labeling models for capturing label dependencies ([38, 35]).

In this research, we investigate two neural sequence taggers for our Ezafe marking problem. The first sequence tagger shares the same encoder as the encoder in our NMT approach but does not need a decoder since each input is synced with an output.

The architecture of the second neural sequence labeling includes an encoder and a CRF layer at the end for considering the dependencies between tags. The encoder itself is comprised of two layers of bi-directional LSTM cells. For the two above-mentioned models, the input of this encoder at each time is a concatenation of a word and its features.

To investigate the performance of our neural sequence labeling approaches, we compared it with a Conditional Random Field (CRF) model proposed by [6].

## 5-3- Ezafe Marking Using NMT approach

Our second approach is to employ the Neural Machine Translation model to tackle the problem of Ezafe recognition. Neural machine translation is an emerging approach to machine translation that has been proposed by [39], [40] and [41]. Unlike the traditional phrase-based translation model such as [42] which consists of many small sub-components that are tuned separately, NMT attempts to build and train a single, large neural network that reads a sentence and results in a correct translation at the output. Most of the proposed neural machine translation models belong to a family of encoder–decoders [40]; [41] with an encoder and a decoder for each language, or employ a language-specific encoder applied to each sentence whose outputs are then compared [43]. An encoder neural network reads and encodes a source sentence into a fixed-length vector. In the next step, a decoder outputs a translation from the encoded vector. The whole encoder-decoder system which consists of the encoder and the decoder for a language pair is jointly trained to maximize the probability of a correct translation given a source sentence. The most common approach for encoder and decoder is to use an RNN, but it should be noted that other architectures such as a hybrid of an RNN and a de-convolutional neural network can be used. [44].

Table 2 – Selection of features

| Feature set | Feature set 1 | Feature set 2 | Feature set 3 | Feature set 4 | Feature set 5 | Feature set 6 |
|---|---|---|---|---|---|---|
| # of features | 2 | 3 | 49 | 50 | 51 | 52 |
| Word conditional probability P(Ezafe\|word) | ■ | ■ | ■ | ■ | ■ | ■ |
| POS | ■ | ■ | ■ | ■ | ■ | ■ |
| POS conditional probability P(Ezafe\|POS) | | ■ | ■ | ■ | ■ | ■ |
| Ending letter (binary factors) | | | ■ | ■ | ■ | ■ |
| Next POS | | | | ■ | ■ | ■ |
| Previous POS | | | | | ■ | ■ |
| Second next POS | | | | | | ■ |

Our NMT model is comprised of an encoder and a decoder, each one contains two layers of LSTM cells. Moreover, the attention model used in the decoder is global attention. Furthermore, we used a random vector as the embedding of the Ezafe marker and constructed the word embedding of the word+Ezafe marker by summation of the embedding of the word and the embedding of the Ezafe marker.

To investigate the performance of our approach, we run two baselines to compare them with neural machine translation. The first baseline is the research done by [6], which proposed a phrase-based SMT approach. For the second baseline, we developed a Factored-based SMT (FB-SMT) approach.

## 6- Experiments and Results

In order to evaluate the method, as the first step an experimental setup should be provided including preparation of training and test corpus. Then we should select effective evaluation measures to accurately evaluate the results. In the next step, we examine six baseline experiments. In baseline experiments 7 and 8 we investigate two competitor approaches that were investigated by [6]. Our approach will be described in experiments 1 and 2 in which the factored-based and neural machine translation models will be investigated. In the next subsections, we will describe the experimental setup and the experiments in detail.

### *A. Experimental Setup*

For investigating the performance of our algorithms, an evaluation framework is required. The framework is comprised of an evaluation corpus along with evaluation measures. In the following subsections, we will thoroughly describe these elements.

#### *Evaluation Corpus:*

In this research, we have used the Bijankhan corpus that is gathered from daily news and common texts ([45, 46]. This corpus contains about 10 million tagged words and covers 4300 different subjects. The words in the corpus have been marked by a tag set containing 550 tags based on a hierarchical order, with more fine-grained POS tags like 'noun-plural-subj'. About 23% of words in the corpus have Ezafe marker tags [1]. The corpus is freely available on the Web for research purposes[1].

Fig. 2 shows the number of Ezafe markers in a sentence (normalized by total words), versus sentence length in Bijankhan corpus. This plot shows that for short sentences, the number of Ezafe markers in a sentence increases with sentence length, whereas in long sentences the average number of Ezafe markers approximately remains constant and does not increase.

In order to unify the character encoding for the next steps, a preprocessing step was applied to the corpus [47]. Furthermore, since determining the border of sentences in our research is of great importance, so the punctuation marks with different character encodings in the corpus were also mapped into standard UTF-8 encodings.

#### *Evaluation Measures:*

Ezafe recognition is indeed a binary classification problem. Precision and Recall are the ordinary measures in this type of classification, which are the measures of exactness and completeness respectively. As a combined measure that assesses precision/recall tradeoff, we have used F-measure (harmonic mean), a parameterized E-measure that equally weights precision and recall.

[1] https://dbrg.ut.ac.ir/span-design/

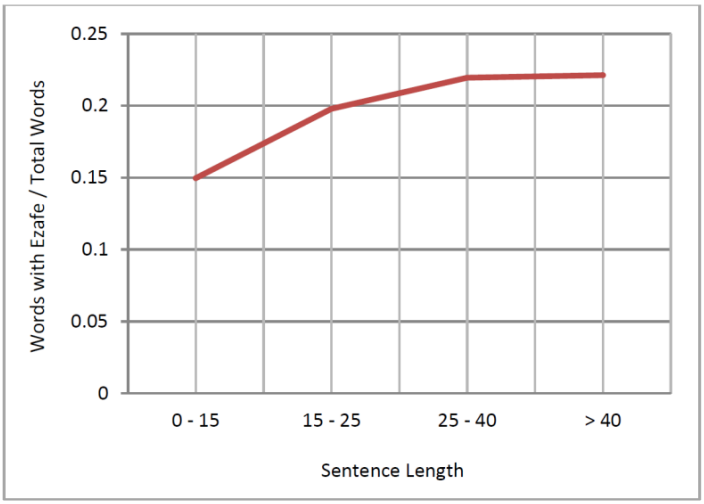


Fig. 2: Percentage of words having Ezafe marker versus Sentence length in the corpus

In our experiments, we have also calculated F0.5 to apply more weight to Precision. This measure can show us how we can deal with the Ezafe tagging problem if we need to emphasize more on Precision rather than Recall. The reason behind this selection is that, in the task of Ezafe recognition, precision is of higher importance with respect to recall. This general aim at high precision has been verified by [48] for evaluating a grammar checker system, and also was in line with Bernth's observations on end-user valuations, in which satisfaction was specified as high precision, i.e. few false recalls, even at a remarkable loss of recall [49]. In Bernth's experiment, even though users expect a proofing tool to find as many errors as possible, they prefer easing up on this expectation if the proportion of correct error flagging is relatively high.

Another measure that can be used in this binary classification problem is the Mathews Correlation Coefficient (MCC). This measure indicates the quality of the classifier for binary class problems especially when two classes are of very different sizes and so there is a class imbalance problem [50]:

$$Mcc = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \qquad (1)$$

where TP, TN, FP, and FN are the true positive rate, true negative rate, false positive rate, and false negative rate, respectively. The Matthews correlation coefficient is often used as a measure of the quality of 2-class binary classifications. It is generally regarded as a balanced measure that can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications. MCC ranges from -1 to 1, where -1 corresponds to inverse classification, zero corresponds to average classification performance and +1 represents perfect classification.

We have also considered two other measures that can be useful in the calculation of accuracy:

$$ezafe\_presence\_accuracy = \frac{TP}{TP + FN} \qquad (2)$$

$$ezafe\_absence\_accuracy = \frac{TN}{TN + FP} \qquad (3)$$

Note that Eq. 2 is the same as the Recall equation. The total average can be calculated using a weighted average of the two above-mentioned equations. In calculating the total weighted average, the weighting factor for Eq. 2 is the percentage of the words with the Ezafe marker in the test corpus which is 18%, while the weighting factor for Eq. 3 is the percentage of the words without the Ezafe marker in test corpus which is 82%. As a result, the weighted accuracy is presented as the final score. This is the calculation that has previously been done by [6], and so the results can be compared with each other.

### *B. Baseline Experiments*

There should always be a simple baseline besides examinations to assess the efficiency of new approaches. This could be random assignment of Ezafe to words that can carry Ezafe markers, assignments based on the frequency of words having Ezafe markers in the training set, etc. So, at the first step in the experiments, we investigated nine types of taggers as baselines. The first six baseline experiments are classic taggers that are based on conditional probabilities of words and/or POS tags.

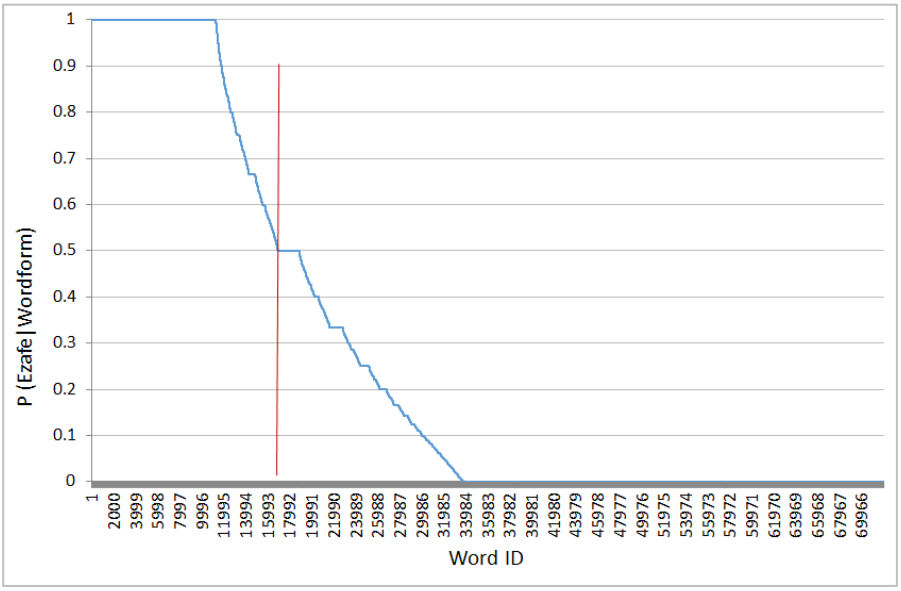


Fig. 3: Conditional probability of Ezafe appearance vs. word ID

Experiments 7 and 8 are Conditional Random Fields (CRF) and monotone SMT approaches that were previously examined by [6]. Moreover, another baseline experiment (experiment 9) was performed based on the Factored-based Machine Translation model. Our approaches are presented in experiments 11 and 12 that incorporate neural sequence tagger and neural machine translation, respectively.
In the following subsections, these approaches will be explained in more detail.

### *Baseline1: Using Word form*

The lexical unit of a word has a great effect on the recognition of Ezafe tags. So, in the first experiment for deriving a baseline, we calculated the conditional probability *P(Ezafe|wordform)*. We counted the number of times each word with the Ezafe marker appeared in the training set. Then, all words were sorted based on their probability of taking the Ezafe marker. In this stage, we need to set a threshold in such a way that all of the words below that point can take the Ezafe marker.

The reason for selecting a threshold of 15% in the graph is as follows: The formula for the Naïve Bayes classifier can be described as:

$$R_1 = \{f(x|w_1)P(w_1) > f(x|w_2)P(w_2)\} \rightarrow R_1 = \{f(x|w_1) > f(x|w_2)\} \quad (4)$$

$$R_2 = \{f(x|w_2)P(w_2) > f(x|w_1)P(w_1)\} \rightarrow R_2 = \{f(x|w_2) > f(x|w_1)\} \quad (5)$$

In the above equation, w1 and w2 are class one and class two respectively, and x is the observation which to be classified in one of these two classes. R1 and R2 are the domain space for the mentioned classes.

In our Ezafe recognition problem, there are two classes of words with the Ezafe marker and words without the Ezafe marker. Based on these two classes, we can calculate the conditional probability of a word form with Ezafe marker:

$$P(Ezafe|word\ form) \propto P(word\ form|Ezafe)P(Ezafe) = \frac{\#word\ forms\ with\ Ezafe}{\#\ word\ with\ Ezafe} \times \frac{\#\ word\ with\ Ezafe}{\#\ Total\ words\ in\ corpus} = \frac{\#word\ forms\ with\ Ezafe}{\#\ \ Total\ words\ in\ corpus} \quad (6)$$

Since we deal with a binary classification problem, we have just two classes in which their sum of probability should become 1.

$$P(Ezafe|word\ form) + P(\sim Ezafe|wordform) = 1 \quad (7)$$

So, for assigning a word form to the Ezafe marker class, it is just enough that:

$$P(Ezafe|word\ form) = \frac{\#\ word\ forms\ with\ Ezafe}{\#\ \ total\ words\ in\ corpus} > 0.5 \quad (8)$$

By investigating a line of probability 0.5 as shown in Fig. 3, we can see that selecting a threshold of 15% of word forms is a good selection for threshold. So, we selected the top 15% of the most probable words in training data that occur with Ezafe marker which *P(Ezafe|wordform)≥Threshold=0.5*, and then used them to mark the same words in the test corpus. Words encountered

in the test set that appear in the top 15% list of the training set were tagged with Ezafe marker. The results are shown in Table 3 named as the " Baseline with word form" approach, which shows a precision of 90.7% and an F-measure value equals to 46.17.

### *Baseline 2: Incorporating Wordform+POS tags*

Part of Speech tags can demonstrate more detailed features of a word, so they can be used to better recognize the presence of Ezafe tags in Persian Text. It has been proven that POS tags are good features for recognizing Ezafe markers. It is because the presence of Ezafe in a word greatly depends on the grammatical features of the word. So, part of speech information can help us to improve Ezafe recognition algorithms.

In this experiment, we calculated the conditional probability *P(Ezafe|wordform, POS tag)*. The results of this approach have been indicated in Table 3 as the 'word form + POS tags' approach, and show a better performance with respect to the previous experiment. The result shows a precision of 93.43% and the F-measure value equals to 54.48.

### *Baseline 3: Using POS tags with FPS*

In the third experiment, the role of POS tags in predicting the Ezafe marker was examined. We calculated the conditional probability *P(Ezafe|POS tag)*. Then we tagged all the words in the test corpus based on a Fitness Proportionate Selection (FPS). In FPS (also known as roulette wheel selection), individuals are given a probability of being selected that is directly proportionate to their fitness [51].

Fig. 4 shows the probability of POS tags having Ezafe marker. The accuracy and other measures of this experiment are shown in Table 3 as the 'POS tags with FPS' approach. The result shows a precision of 34.88%, while recall is 40.63%, so the F-measure value equals to 37.54.

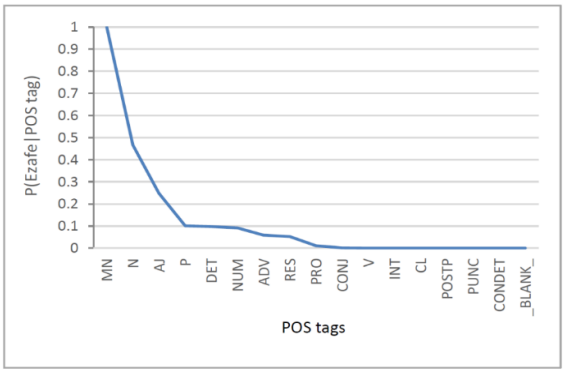


Fig. 4: Conditional probability of Ezafe appearance vs. POS tags

### *Baseline 4: Word form with FPS*

In this experiment, we examined the role of all word forms in predicting the Ezafe marker. At first, we calculated the conditional probability of *P(Ezafe|wordform)*. In the next step, all the words in the test corpus were tagged based on a Fitness Proportionate Selection method. The results of the accuracy of this approach are shown in Table 3 as the 'word form with FPS' approach. The result shows a precision of 66.62%, recall of 66.64% and the F-measure value equals to 66.63.

### *Baseline 5: word form + POS tag with FPS*

In this experiment, the conditional probability *P(Ezafe|*wordform, *POStag)* was derived. Then we tagged all the words in the test corpus based on the value of this conditional probability on a Fitness Proportionate Selection basis. The results of this approach are shown in Table 3 marked as the 'wordform+POS tag with FPS' approach. In this experiment, we achieved a precision and recall of 68.02% and the F-measure value equals to 68.02 as shown in Table 3.

### *Baseline 6: Binary Classifier*

In this experiment, we developed a simple binary classifier that used a window around a word to predict the existence of Ezafe marker. This is a basic baseline system for tasks like WSD. At first, we selected a window size of 3 and the features of the classifier were as follows:

- Current word
- Ending letter of the current word
- POS tag of the current word
- POS tag of previous adjacent word
- POS tag of next adjacent word

All the words in the corpus were classified by this classifier with a 10-fold cross-validation approach and the results show a precision of 84.60%, recall of 94.10%, and F-measure value equals to 89.10. In the next step, we examined the effect of window size on the performance of the classifier. We selected a window size of 5 which means we considered the POS tag of two words before and two words after the current word as well as the current word itself, the ending letter of the current word, and the POS tag of the current word. The result shows a precision of 85.13%, recall of 94.20%, and F-measure value equals to 89.44 which is a little better than the results of window size 3. The results of these two approaches are marked in Table 3 as 'Window Size 3' and 'Window Size 5'.

### *Baseline 7: Conditional Random Field Model*

The next experiment was examined based on Conditional Random Field (CRF) which is a framework for building probabilistic models to segment and label sequence data.

Table 3: Baseline approaches

| Baseline Experiments | Features | Precision | Recall | Accuracy | MCC | $F_1$ | $F_{0.5}$ |
|---|---|---|---|---|---|---|---|
| Baseline #1 | Word form | 90.7 | 30.96 | 87.57 | 48.50 | 46.17 | 65.44 |
| Baseline #2 | Word + POS tags | 93.43 | 38.44 | 88.92 | 54.27 | 54.48 | 72.64 |
| Baseline #3 | POS tags with FPS | 34.88 | 40.63 | 78.58 | 23.11 | 37.54 | 35.89 |
| Baseline #4 | Word form with FPS | 66.62 | 66.64 | 86.07 | 66.63 | 66.63 | 66.62 |
| Baseline #5 | Word form +POS with FPS | 68.02 | 68.02 | 86.82 | 68.02 | 68.02 | 68.02 |
| Baseline #6 | Binary classifier Window size 3 | 84.60 | 94.10 | 94.8 | 85.9 | 89.1 | 86.3 |
| | Binary classifier Window size 5 | 85.13 | 94.20 | 95.0 | 86.4 | 89.44 | 86.8 |
| Baseline #7 | CRF (Asghari et al, 2014) | 94.81 | 96.64 | 98.0 | 94.4 | 95.72 | 95.15 |
| | CRF (with feature set 6) | 95.05 | **96.85** | **98.16** | **94.76** | **95.94** | 95.4 |
| Baseline #8 | PB-SMT (Asghari et al, 2014) | 82.42 | 75.91 | 86.82 | 77.81 | 79.03 | 81.03 |
| Baseline #9 | FB-SMT (with feature set 6) | **95.94** | 94.47 | 97.22 | 94.32 | 95.20 | **95.64** |

CRFs are a type of discriminative undirected probabilistic graphical models.

The definition of CRF on observations X and random variables Y would be as follows: Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then *(X,Y)* is a conditional random field when the random variable $Y_v$, conditioned on X, obey the Markov property with respect to the graph:

$p(Y_v|X, Y_w, w \neq v) = p(Y_v|X, Y_w, w \sim v)$

where $w \sim v$ means that w and v are neighbors in G.

The experiment was performed based on 10-fold cross-validation. According to the previous research by [6], we set the window size to 5 for achieving the best performance. Based on this experiment, we achieved a precision of 94.81%, recall of 96.64% and F-measure value equals to 95.72.

We also investigated the CRF method of [6] with feature set 6. A precision of 95.05%, recall of 96.85%, and F-measure value equals to 95.94 was achieved. The difference is that instead of post processing Persian-specific features, we used all the features as a whole into the model and the results show better performance with respect to the [6].

### *Baseline 8: Monotone SMT approach*

The Ezafe recognition problem can be considered as a translation problem. The original training text without the Ezafe marker can be used as the source language, and the tagged text can be mentioned as the target language. So, we can use these parallel corpora as training data into a machine translation system. In the testing phase, the text without Ezafe markers would be converted into text with Ezafe markers.

In this experiment, we re-examined the approach of [6] on our corpus to compare it with our new approach. So, a phrase-based Statistical Machine Translation (PB-SMT) algorithm was incorporated with a distortion limit set to zero. Table 3 shows the result of the simple monotone translation approach with precision rates of 82.42%, recall as 75.91, and F-measure of 79.03.

### *Baseline 9: Factor-based SMT approach*

In this experiment, we have exploited Factor-based SMT model. The Statistical Machine Translation system only considers the surface word forms of sentences and does not include the linguistic knowledge of the languages. So, its performance is poor for dissimilar language pairs when compared to similar language pairs. The factored model was introduced as an extension of phrase-based SMT to reduce the problems of the inability to handle linguistic description beyond surface forms [52]. In a factored model, the system no longer translates words. Instead, each word is represented by a vector of factors that can contain the surface form, but also the lemma, word class, morphological characteristics, or any other information relevant to translation. Factored models can employ various types of additional information to improve translation quality between language pairs. A word in the FB-SMT framework is not a simple token; instead, it is a vector of factors representing different levels of annotation [52]. As in phrase-based translation, the main source of data for training factored models is a parallel corpus.

Table 4: Comparison of various approaches to Persian Ezafe recognition

| No | Approaches | Precision | Recall | Accuracy | MCC | F1 | F0.5 |
|---|---|---|---|---|---|---|---|
| 1 | Neural sequence labeling (with feature set 6) | 95.74 | **96.85** | **98.27** | **95.16** | **96.29** | **95.96** |
| 2 | Neural Sequence Labeling – BILSTM+CRF (with feature set 6) | **95.78** | 96.47 | 98.25 | 95.00 | 96.13 | 95.92 |
| 3 | NMT (with feature set 6) | 95.41 | 95.91 | 98.04 | 94.40 | 95.66 | 95.51 |

While phrase-based translation models usually memorize local translation literally and make independent assumptions between phrases which makes the model not to be in sentence level, FB-SMT models provide better generalization and richer structures [53]. In this experiment, we have examined the FB-SMT model as an approach to Ezafe recognition, and various features were used as factors into the model.

The result of FB-SMT is shown in Table 3. We selected feature set 6 of Table 2 since it has resulted in the best performance with respect to the other feature sets. The result shows a precision of 95.94% and an F-measure value equal to 95.20 with a total accuracy of 97.22%.

### *C. Experiments with Neural Approaches*

#### *Experiment 1: Neural Sequence Labeling*

In this experiment, we have examined two neural sequence tagging models to be employed into our Ezafe recognition problem. For the first tagger, we utilized the default sequence tagger of OpenNMT Torch which is an open-source initiative for neural machine translation and sequence modeling [54]. For the second tagger, we investigated a more advanced model that incorporates a Bi-LSTM-based encoder into a CRF layer for capturing label dependencies and implemented it by the use of the OpenNMT-tf. The input of this encoder at each time is a concatenation of a word and its features. We used the Bijankhan corpus for training the models and performed a 5-fold cross-validation method to get significant results.

The results of the Neural Sequence Labeling approach have been depicted in Table 4. According to the table, in the first model, we have reached improved results with respect to baseline approaches with precision rates of 95.74%, recall at 96.85% and F1 equals to 96.29. The Neural Sequence Tagger method shows improvements in F1 and F0.5 measure in comparison with the CRF approach by 1.003 and 0.56 respectively.

Furthermore, in the second model, we investigated a BILSTM-CRF model with the same feature set. It indicates lower rates in comparison to the first one with respect to recall (96.47%) and F-measure (96.13), but its precision improves the first experiment by 95.78%.

#### *Experiment 2: Neural Machine Translation*

In this experiment, we investigated the NMT model for the Ezafe recognition task. Our NMT model includes an encoder and a decoder, each one contains two layers of LSTM cells. For this experiment, we used OpenNMT and provided it with pre-trained word embeddings. We used a random vector as the embedding vector of the Ezafe marker and constructed the word embedding of the Word+Ezafe marker by the summation of the embedding of the word and the embedding of the Ezafe marker. The input of this encoder at each time is a concatenation of a word and its features. We used Bijankhan corpus with the feature set 6 as their features for the input of this tagger, with a 5-fold cross-validation for getting better results.

To investigate the performance of our approach, we run two baseline models for comparing them with neural machine translation. The first baseline is a PB-SMT model proposed by [6] and for the second one, we developed a Factored-based SMT (FB-SMT) approach as presented in Table 3.

Table 4 shows the result of the Neural Machine Translation approach with incorporation of all the features in feature set 6 which results in precision equals to 95.41%, recall equals to 95.91% and F-measure reaches 95.66.

## 7- Discussion

We investigated eleven experiments to evaluate the performance of different approaches to Ezafe marking. The first nine experiments depicted in Table 3 were the baselines, in which CRF and PB-SMT (experiments 7 and 8) are the two competitor approaches that previously investigated by [6]. By the use of feature set 6, the CRF approach can perform better results than that of [6]. This baseline experiment shows the effectiveness of the syntactic features used in our investigations. Moreover, the FB-SMT approach achieved the best results when dealing with $F_{0.5}$ and precision as well.

Our approaches have been presented in experiments 1 and 2, in which two Neural Sequence Labeling methods and an

NMT model were examined. It is worth mentioning that for a fair comparison, we implemented the models with the same features as of the FB-SMT and CRF baseline models. Since CRF is the best approach of previous work investigated by [6], we selected it as the best baseline for comparing to our approaches. The results show that the neural Sequence Labeling approach can perform better than CRF method in F1 and $F_{0.5}$ by 0.35 and 0.56 respectively, but the recall rate of the two approaches is the same.

Table 5: Flase Positive and False Negative matrix of CRF and Neural Sequence Labeling approach

| Recognition Algorithm | FN | FP |
|---|---|---|
| CRF | 71159 | 109787 |
| Neural Sequence Labeling | 70010 | 95651 |

By investigating the behavior of the neural models, we have studied the False Positive and False Negative rates versus CRF as the best baseline. Table 5 shows the number of false positive and false negative cases in CRF and Neural Sequence Labeling as the two competitors. As can be seen in the table, there is not a big difference between the numbers of FNs in the table; we can say that the FN rates of the two methods are approximately the same. So we focus on the FP rate in more detail to evaluate the differences. By investigating the various POS tags in FP cases, the most important differences between the two methods belong to N-N and N-Adj POSE tags. So we conclude that the Neural Sequence Labeling approach can achieve better performance when encountering N-N and N-Adj cases.

Table 6: Some examples that shows the performance of Neural Sequence Labeling over CRF

| No | Example | True Case | Neural Sequence Labeling | CRF |
|---|---|---|---|---|
| 1 | آنها استفاده از روش تک کتابی را در تدریس (N) ترک (N) کرده اند.<br>?ânhâ Estefâdeh æz ræveʃe tæk ketâbi râ dær tædris-EZ tærk kærdeh ænd. | N-N | TN | FP |
| 2 | معلم در ترغیب کودکان مدرسه (N) مسئولیت (N) سنگینی در ایجاد رغبت مطالعه دارد<br>Moællem dær tærqibe kudækâne mædrese-EZ mæsouliate sængini dær ijâde ræqbæte motâlele dâræd. | N-N | TN | FP |
| 3 | ما مدعی هستیم که الجزیره یک بنگاه خبری ست، نه سازمانی (N) ایدئولوژیک (ADJ)<br>Mâ modæie hæstim ke æljæzireh yek bongâhe khæbæri æst næ sâzmâni-EZ ideologic | N-ADJ | TN | FP |
| 4 | کارشناسان مسائل رسانه‌ای این نقش را به مراتب (N) مهمتر (ADJ) و تأثیرگذارتر در جنگ خلیج فارس می‌دانند.<br>Kârʃenâsâne mæsâlele ræsâneiy in næqʃ râ be mærâteb-EZ mohemtær væ tasir gozârtær dær jænge khælije fârs midânænd. | N-ADJ | TN | FP |

Some examples that show the advantage of Neural Sequence Labeling with respect to CRF approach are depicted in Table 6. The examples show that Neural Sequence Labeling usually performs well in the case of N-ADJ or N-N.

## 8- Conclusion

In this study, some experiments on Persian Ezafe recognition were conducted to test the impact of neural approaches in automatic Ezafe recognition. The baseline experiments were designed based on a combination of features such as word forms, POS tags, and ending letter of each word to obtain a baseline for comparing to our new approaches. The results partially confirmed the claim that there is poor accuracy by using simple baseline approaches, while CRF approach and FB-SMT performed well among all of the baselines.

The first contribution of this study is to use neural sequence labeling models, in which we exploited some lexical and grammatical Persian-specific features as factors into the model. At first, we used a Neural Sequence Labeling model. The results show a better performance with respect to the baseline experiments. We also investigated a BILSTM+CRF sequence labeling model which although shows a better precision rate, it cannot perform better than the first model in the case of F1 measure. The reason might be that we have just two labels and there is not any special dependency between the labels (with or without Ezafe marker).

The second contribution was based on Neural Machine Translation along with feature set 6 as features into the model. The performance of the NMT approach outperforms other MT approaches in the baselines by F1. But still FB-SMT model has better performance in the case of Precision and F0.5. In comparing NMT to Neural Sequence labeling models, both Neural Sequence Labeling models outperform the NMT approach.

As a result, adding various Persian-specific features to the Neural Sequence labeling algorithm resulted in a significant improvement in precision and recall with respect to CRF approach. Moreover, our approach outperforms the CRF method in the case of F0.5 measure in which the precision is more important than recall.

A research line that can be proposed for future studies is tackling the problem of Ezafe recognition as a spell-checking problem. Suppose the words that should take Ezafe markers in the original text are misspelled words. So, the problem of Ezafe recognition can be defined as a spell-checking problem; finding the words that are incorrectly written without Ezafe tags, and correcting them to the words with Ezafe marker. The output of the system can be regarded as the corrected text. One more suggestion for future work is to implement a rule-based approach by

incorporating error-driven Transformation Based Learning or TBL [55], in which a sequence of rules is accepted to be applied to the corpus that leads to the most improvement in error reduction in the text. In TBL, the rules are learned iteratively and must be applied in an iterative fashion for retagging. So we may require a rule-ordering mechanism; Rules become increasingly specific as we go down the sequence. More specific rules cover just a few cases. In TBL, we should also set a stopping criterion; learning is stopped when we reach an error rate lower than a predefined threshold. The advantage of TBL is that, unlike statistical methods, allows making more sense of rules and their actions.

Another research line that can be proposed for future studies is exploiting word embeddings to solve the problem of Ezafe recognition.

## Acknowledgments

We gratefully acknowledge the help and support provided by the members of the Natural Language Processing Lab, University of Tehran. The authors also wish to express their gratitude to Mr. Amin Mansouri, Behzad Mirzaabaee, Nima Hemmati, and Arezou Hatefi for their valuable assistance. Without their help, this work would have not been possible.

## References

[1] Bijankhan, M., Sheykhzadegan, J., Bahrani, M., & Ghayoomi, M. (2011). Lessons from building a Persian written corpus: Peykare. Language resources and evaluation, 45(2), 143-164.

[2] Bahaadini, S., Sameti, H., & Khorram, S. (2011, September). Implementation and evaluation of statistical parametric speech synthesis methods for the Persian language. In Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on (pp. 1-6). IEEE.

[3] Oskouipour, N. (2011). Converting Text to phoneme stream with the ability to recognizing ezafe marker and homographs applied to Persian speech synthesis. Msc. Thesis, Sharif University of Technology, Iran.

[4] Maleki, J., Yaesoubi, M., & Ahrenberg, L. (2009, July). Applying Finite State Morphology to Conversion Between Roman and Perso-Arabic Writing Systems.Proceedings of the 2009 conference on Finite-State Methods and Natural Language Processing: Post-proceedings of the 7th International Workshop FSMNLP 2008 (pp. 215-223).

[5] Nourian, A, Rasooli, M. S., Imany, M., and Faili, H., (2015) On the Importance of Ezafe Construction in Persian Parsing, The 53rd Annual Meeting of the Association for Computational Linguistics (ACL) and the 7h International Joint Conference on Natural Language Processing (IJCNLP), Beijing, China, July 2015., Volume 2: Short Papers: 877.

[6] Asghari, H., Maleki, J., & Faili, H. (2014). A Probabilistic Approach to Persian Ezafe Recognition. 14th Conference of the European Chapter of the Association for Computational Linguistics(EACL 2014 ), 138. 26–30 April 2014, Gothenburg, Sweden.

[7] Noferesti, S., and Shamsfard, M., (2014). A Hybrid Algorithm for Recognizing the Position of Ezafe Constructions in Persian Texts.International Journal of Artificial Intelligence and Interactive Multimedia (IJIMAI) 2(6): 17-25 (2014).

[8] Isapour, S., Homayounpour, M. M., and Bijankhan, M. (2007). Identification of ezafe location in Persian language with Probabilistic Context Free Grammar, 13th Computer Association Conference, Kish Island, Iran.

[9] Farghaly, A., (2004). Computer Processing of Arabic Script-based Languages: Current State and Future Directions. Workshop on Computational Approaches to Arabic Script-based Languages, COLING 2004, University of Geneva, Geneva, Switzerland, August 28, 2004.

[10] Asgari, E., & Mofrad, M. R. (2016). Comparing Fifty Natural Languages and Twelve Genetic Languages Using Word Embedding Language Divergence (WELD) as a Quantitative Measure of Language Distance. arXiv preprint arXiv:1604.08561.

[11] Marno, H., Langus, A., Omidbeigi, M., Asaadi, S., Seyed-Allaei, S., & Nespor, M. (2015). A new perspective on word order preferences: the availability of a lexicon triggers the use of SVO word order. Frontiers in Psychology, 6.

[12] Moghaddam, M. D. (2001). Word order typology of Iranian languages. The Journal of Humanities of the Islamic Republic of Iran.–2001 (Spring), 8(2), 17-23.

[13] Parsafar P. (2010). Syntax, Morphology, and Semantics of Ezafe. Iranian Studies [serial online]. December 2010; 43 (5): 637-666. Available in Academic Search Complete, Ipswich, MA.

[14] Bögel, T., Butt, M., and Sulger, S., (2008). Urdu ezafe and the morphology-syntax interface. Proceedings of LFG08 (2008). CSLI Publications Stanford.

[15] Estaji, A., and Jahangiri, N. (2006). The origin of kasre ezafe in Persian language. Journal of Persian language and literature, Vol. 47, pp. 69-82, Isfahan University, Iran.

[16] Samvelian, P. (2007). 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, 339-361.

[17] Megerdoomian, K. (2000). A computational analysis of the Persian noun phrase. Memoranda in Computer and Cognitive Science MCCS-00-321, Computing Research Lab, New Mexico State University.

[18] Bijankhan, M. (2005). A feasibility study on Ezafe Domain Analysis based on pattern matching method. Published by Research Institute on Culture, Art, and Communication, Tehran, Iran.

[19] Zahedi, M. (1998). Design and Implementation of an Intelligent Program for Recognizing Short Vowels in Persian Text. Msc. Thesis, University of Tehran, Iran.

[20] Mavvaji, V., and Eslami, M., (2012). Converting Persian Text to Phoneme Stream Based on a Syntactic Analyser. The first international conference on Persian text and speech, September 5,6, 2012, Semnan, Iran.

[21] Razi, B., and Eshqi, M., (2012). Design of a POS tagger for Persian speech based on Neural Networks, 20th Conference on Electrical Engineering, 15-17 May 2012, Tehran, Iran.

[22] Chen, X., Qiu, X., Zhu, C., Liu, P., & Huang, X. (2015). Long short-term memory neural networks for Chinese word segmentation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1197-1206).

[23] Ma, X., & Hovy, E. (2016). End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1064-1074)

[24] Cuong, N. V., Ye, N., Lee, W. S., & Chieu, H. L. (2014). Conditional random field with high-order dependencies for sequence labeling and segmentation. The Journal of Machine Learning Research, 15(1), 981-1009.

[25] Farghaly, A., & Shaalan, K. (2009). Arabic natural language processing: Challenges and solutions. ACM Transactions on Asian Language Information Processing (TALIP), 8(4), 14.

[26] Elshafei, M., Al-Muhtaseb, H., & Al-Ghamdi, M. (2006 a). Machine Generation of Arabic Diacritical Marks. MLMTA, 2006, 128-133.

[27] Elshafei, M., Al-Muhtaseb, H., & Al-Ghamdi, M. (2006 b). Statistical methods for automatic diacritization of Arabic text. In The Saudi 18th National Computer Conference. Riyadh (Vol. 18, pp. 301-306).

[28] Diab, M., Ghoneim, M., & Habash, N. (2007, September). Arabic diacritization in the context of statistical machine translation. In Proceedings of MT-Summit.

[29] Habash, N., & Rambow, O. (2007, April). Arabic diacritization through full morphological tagging. In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2007); Companion Volume, pp. 53-56, Association for Computational Linguistics.

[30] Belinkov, Y., & Glass, J. (2015). Arabic diacritization with recurrent neural networks. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 2281-2285).

[31] Abandah, G. A., Graves, A., Al-Shagoor, B., Arabiyat, A., Jamour, F., & Al-Taee, M. (2015). Automatic diacritization of Arabic text using recurrent neural networks. International Journal on Document Analysis and Recognition (IJDAR), 18(2), 183-197.

[32] De Mareüil, P. B., Adda-Decker, M., & Gendner, V. (2003). Liaisons in French: a corpus-based study using morpho-syntactic information. In Proc. of the 15th International Congress of Phonetic Sciences.

[33] Larson, R.K. (2009), Chinese as a reverse Ezafe language. Yuyanxue Luncong, Journal of Linguistics, 39: 30-85. Peking University.

[34] Ling, W., Dyer, C., Black, A. W., Trancoso, I., Fermandez, R., Amir, S., ... & Luis, T. (2015). Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (pp. 1520-1530).

[35] Peters, M., Ammar, W., Bhagavatula, C., & Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 1756-1765).

[36] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[37] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4), 541-551.

[38] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural Architectures for Named Entity Recognition. In Proceedings of NAACL-HLT (pp. 260-270).

[39] Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 48–54, Stroudsburg, PA, USA. Association for Computational Linguistics.

[40] Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1700–1709. Association for Computational Linguistics.

[41] Sutskever, I., Vinyals, O., and Le, Q. (2014). Sequence to sequence learning with neural networks. In Advances in Neural Information Processing Systems (NIPS 2014).

[42] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014a). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[43] Hermann, K. M., & Blunsom, P. (2014). Multilingual models for compositional distributed semantics. arXiv preprint arXiv:1404.4641.

[44] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[45] Bijankhan, M. (2004). The Role of the Corpus in Writing a Grammar: An Introduction to a Software, Iranian Journal of Linguistics, Vol. 19, No. 2, fall and winter 2004.

[46] Amiri, H., Hojjat, H., & Oroumchian, F. (2007). Investigation on a feasible corpus for Persian POS tagging. In Proceedings of the 12th International CSI Computer Conference (CSICC), 2007.

[47] Mohtaj, Salar, Behnam Roshanfekr, Atefeh Zafarian, Habibollah Asghari, (2018), Parsivar: A Language Processing Toolkit for Persian, 11th edition of the Language Resources and Evaluation Conference (LREC 2018), 7-12 May 2018, Miyazaki (Japan).

[48] Arppe, A. (2000). Developing a grammar checker for Swedish. In Proceedings of NODALIDA (Vol. 99, pp. 13-27).

[49] Bernth, A. (1997, March). EasyEnglish: a tool for improving document quality. In Proceedings of the fifth conference on applied natural language processing (pp. 159-165). Association for Computational Linguistics.

[50] Powers, D.M.W., (2011). Evaluation: from Precision, Recall, and F-measure to ROC, Informedness, Markedness, and Correlation. Journal of Machine Learning Technologies, 2(1), 37-63.

[51] Bäck, T. (1996). Evolutionary algorithms in theory and practice. Oxford University Press.

[52] Koehn, P., & Hoang, H. (2007, June). Factored Translation Models. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, In EMNLP-CoNLL 2007, pp. 868–876, Prague, June 2007. Association for Computational Linguistics.

[53] Feng, Y., Cohn, T., & Du, X. (2014). Factored Markov translation with robust modeling. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning (pp. 151-159).

[54] Klein, G., Kim, Y., Deng, Y., Senellart, J., & Rush, A. (2017). OpenNMT: Open-Source Toolkit for Neural Machine Translation. Proceedings of ACL 2017, System Demonstrations, 67-72.

[55] Brill, E. (1995). Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging. Computational Linguistics, 21(4), 543-565.

[56] Kahnemuyipour, Arsalan (2003). Syntactic categories and Persian stress. Natural Language & Linguistic Theory 21.2: 333-379.

[57] Ghomeshi, J. (1996). Projection and Inflection: A Study of Persian Phrase Structure. Ph.D. Thesis, Graduate Department of Linguistics, University of Toronto.

[58] Doostmohammadi, E., Nassajian, M., & Rahimi, A. (2020, November). Persian Ezafe Recognition Using Transformers and Its Role in Part-Of-Speech Tagging. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 961-971).

[59] Larson, R., & Samiian, V. (2020). The Ezafe construction revisited. Advances in Iranian linguistics, 351, 173.

[60] Ansari, A., Ebrahimian, Z., Toosi, R., & Akhaee, M. A. (2023, May). Persian Ezafeh Recognition using Transformer-Based Models. In 2023 9th International Conference on Web Research (ICWR) (pp. 283-288). IEEE.

[61] Kharsa, R., Elnagar, A., & Yagi, S. (2024). BERT-Based Arabic Diacritization: A state-of-the-art approach for improving text accuracy and pronunciation. Expert Systems with Applications, p. 123416.

[62] Lapointe, M., Kadim, A., & Dliou, A. (2023, November). Literature Review of Automatic Restoration of Arabic Diacritics. In 2023 IEEE International Conference on Advances in Data-Driven Analytics And Intelligent Systems (ADACIS) (pp. 1-5). IEEE.