Recognizing Transliterated English Words in Persian Texts

Ali Hoseinmardy
Computer Engineering Department, Amirkabir University of Technology, Iran
ali.hoseinmardy@aut.ac.ir
Saeedeh Momtazi*
Computer Engineering Department, Amirkabir University of Technology, Iran
momtazi@aut.ac.ir

Received: 28/Jan/2020 Revised: 12/Feb/2020 Accepted: 19/Apr/2020

Abstract

One of the most important problems of text processing systems is the word mismatch problem. This results in limited access to the required information in information retrieval. This problem occurs in analyzing textual data such as news, or low accuracy in text classification and clustering. In this case, if the text-processing engine does not use similar/related words in the same sense, it may not be able to guide you to the appropriate result.

Various statistical techniques have been proposed to bridge the vocabulary gap problem; e.g., if two words are used in similar contexts frequently, they have similar/related meanings. Synonym and similar words, however, are only one of the categories of related words that are expected to be captured by statistical approaches. Another category of related words is the pair of an original word in one language and its transliteration from another language. This kind of related words is common in non-English languages. In non-English texts, instead of using the original word from the target language, the writer may borrow the English word and only transliterate it to the target language. Since this kind of writing style is used in limited texts, the frequency of transliterated words is not as high as original words. As a result, available corpus-based techniques are not able to capture their concept. In this article, we propose two different approaches to overcome this problem: (1) using neural network-based transliteration, (2) using available tools that are used for machine translation/transliteration, such as Google Translate and Behnevis. Our experiments on a dataset, which is provided for this purpose, shows that the combination of the two approaches can detect English words with 89.39% accuracy.

Keywords: Transliteration; Text processing; Words Relation; Neural Network-Based Sequence2Sequence Model; Google Translate; Behnevis

1- Introduction

Searching textual information on the Web has become one of the main usages of the Internet. People use the Internet to find the information they need. For this reason, the intelligence of language processing tools can be much helpful in interacting with computers.

One of the challenges encountered in text processing systems is recognizing related words in a language. Different models have been proposed to address this issue, most notable methods are based on dictionary and statistical co-occurrence of words.

The available methods, however, have not been able to produce good results for new words entered into a language. Transliterated words from dominated languages such as English to other languages are examples of new words in the target languages. Transliterated words are words that are entered in the target language with their vocals. These words are written in the target language as they are pronounced in the source language.

The words "دیپ" (the transliteration of the word "deep" in Persian), "تیبل" (the transliteration of the word "table" in Persian) are examples of this phenomenon in non-English languages which should be captured as related words to "عمیق" and "میز", respectively, which are the correct translations.

The goal of this paper is detecting this type of words in a text and finding a relation between a word in the target language and a transliterated word from another language; e.g., the words "دیپ" and "عمیق" in the above example in Persian. To the best knowledge of the authors, this issue has not been studied in Persian and this is the first attempt in this direction.

The structure of this paper is as follows: In Section 2, we describe word relation, Section 3 describes the transliteration in the Persian language and its challenges. In Section 4, we show the proposed model and the solutions which include 2 different techniques: the seq2seq

* Corresponding Author

model and the tool-based technique. Section 5 represents the evaluation dataset and the results. Finally, Section 6 concludes the paper.

2- Word Relationship

Various methods have been proposed to overcome the word miss-match problem. Dictionary-based and distributional methods are the main techniques used for this goal.

The dictionary-based methods are very simple and its implementation is very easy. The main problem of this approach is the fact that the dictionary vocabulary is constant; i.e., new words in a language, do not exist dictionaries, since updating dictionaries is very costly and time-consuming. Another problem with this method is that it does not consider foreign words. Words transliterated from another language to Persian, which is the target of this research, are not included in the Persian dictionaries.

In the distributional methods, the contexts of the words are used to identify their concept. The advantage of this method compared to the former one is the possibility of adding new vocabulary items at any time by using texts published daily on the Internet. For example, in the Persian language, if we search for the phrase "نرخ ماشین" ("the price of the car"), we expect to receive quite similar results compare to the query "قیمت اتومبیل", which has the same meaning.

One of the problems of this approach is requiring a large amount of data and context for each word. Therefore, although these methods have achieved desirable results in different languages including Persian [7], their success is limited to the words used frequently in those languages.

By the rapid growth of the WEB2 content and the huge amount of information that ordinary people enter into the Internet via social networks, blogs, etc., transliterated foreign words are added daily in non-English texts.

In some cases of transliterated words, as they are used very common in the target language, they can be detected by distributional methods due to their high frequency; e.g., the word "تست" (transliteration of the word "test" from English), which is used by everyone, became a formal translation of the word as well. Therefore, its concept and its similarity to the word "آزمون" can be detected with the statistical methods. For most of the transliterated words, however, it is almost impossible to identify them using these two methods, due to the lower frequency they have.

For example, the word "دیپ" (the transliteration of the word "deep"), which is used in Persian text, should be recognized as a related word to "عمیق", but since this word is not available in the Persian dictionary and its frequency is not high in Persian, we could not find such relation. As a result, we cannot expect similar results when searching for the words "دیپ" and "عمیق". Such a problem motivated us to propose a new solution to overcome this problem.

3- Transliteration

Transliteration means a word with fixed vocals (sound and pronunciation) transfers from one language to another, and each letter is transmitted in the same way as another language. For example, the word "کامپیوتر" is the transliterated word of "computer" in Persian [8].

Transliteration is almost obvious for people who know the both languages. This makes transliteration a personalized task; therefore, it is different from transcription. Because transcription maps the sound of one language to another. There are lots of difficulties in transliteration, like "ق" in Arabic and Persian. Because in English it's not "k" or "g" and maybe something between these two [8].

In the transliteration task between two languages, reconstructing the original form of writing a word in the source language from the transliterated form with no ambiguity is an important issue. Therefore, a character/letter does not need to be always translatable. Another thing is conceivability when several different characters are spelled alike. In this case, there is usually a rule in the original language that decides which characters should be converted when the word is spelled. For instance, if we consider "ک" and "ک" in the Persian language as two different characters, the spelled-out form of the word "کک" in English is "kk". Then, to reconstruct the word to the original language from "kk", it should be known that "k" is written "ک" in the Persian language if it comes first, and if it comes at the end of a character that adheres to the next letter, "k" is written "ک". Therefore, with the basic knowledge of Persian script, we can still reconstruct the original form of the word.

One important issue in transliteration is that in some cases, people use different types of transliteration for a foreign word; e.g., the English word "balcony", is transliterated to "بالکن" and also "بالاکن" in the Persian language.[1]

To summarize, the main limitations of transliteration are as follows:

- There may be more than one transliteration for a word. People may pronounce or write an English word in different ways in Persian, Like "Mouse" in English which can be written like "ماوس" and "موس"

There might be some words that are written in the same way in another language. Like "Test" & "Toast" in English which are written "تست" in Persian.

[1] according to the transliteration style sheet, the second word is wrong, and the first word should be used.

Table 1: Constants of Persian language according to Transliteration style sheet presented by Dr. Mo'in

Constants																						
English Alphabets	B	C	D	F	G	H	J	K	L	M	N	P	Q	R	S	š	T	V	X	Z	ž	'
Persian Alphabets	ب	چ	د	ف	گ	ه‌ح	ج	ک	ل	م	ن	پ	غ‌ق	ر	ث‌س‌ص	ش	ت‌ط	و	خ	ذ‌ز‌ض‌ظ	ژ	ع‌ء

3-1- Transliteration style Sheet

This transliteration style sheet was first used by Dr. Mohammad Mo'in in its famous Encyclopedia in 1971[14] to represent the correct pronunciation of words. Later in 2012, the same model, entitled "General Transcription Style Sheet for Geographical Names of Iran", was sent to all ministries and state. This standard is accepted by the United Nations. [21]

Table 2: Vowels of Persian language according to Transliteration style sheet presented by Dr. Mo'in

Vowels								
Persian Alphabets	ا	اِ	اُ	آ	ئی	ئو	ع‌ء	اَوْ
English Alphabets	a	e	o	ā	i	u	ë'/	ow
Persian Example	ابر	ارم	اردک	آسمان	ایران	بوستان	مسئول	فردوسی
Equivalents	Abr	Eram	Ordak	Āsemān	Irān	Bustān	Mas'ul	Ferdowsi

As can be seen in Tables 1 and 2, the transliteration of Persian to English and the transliteration of English to Persian is based on phonemes and letters. Transliteration is usually used in proper nouns and named entities, such as the particular name of persons, organizations, locations, or events; e.g., "Jonathan", “Apple”, or "Nowruz". Ghayoomi et al. [6] and Bijankhan et. al. [2] provided a comprehensive study on challenges related to the Persian transliteration style.

3-2- Transliteration Challenges

Transliteration has different challenges that we describe in this section. Different dialects in languages: there are several dialects in the languages; e.g., the way American people speak is different from the Canadian people, but both of them speak English.
Different transliteration for a word: Due to phonemes and pronunciations in different languages, a word in one language (for example, the word "اسلام" in Persian) may have several English transliterations ("Islam" or "Eslam"). In this state, we must refer to the original word chosen for the meaning of "اسلام" in English ("Islam"), and count it as the correct word.
Words with letters that are not pronounced: some words have letters that we do not pronounce them; e.g., “خواهر” and “خورشید” in Persian or “knife” and “knee” in English. According to the method used in this project, they will be implicitly solved when processing them in our proposed methods.

4- The Proposed Model

This research aims to provide a tool which receives a Persian text as an input and detects foreign (transliterated) words in the text, Furthermore, it finds the corresponding Persian meaning. For this purpose, the project can be implemented in two steps: (1) detecting if a word is an original word or a transliterated word, and (2) finding the equivalent of the foreign word in Persian; i.e., the Persian meaning of that word.
To this aim, we propose two different approaches which will be described in the next sections:

- Using neural network-based transliteration
- Using available tools, such as Google Translate, and Behnevis.

Figure 1 shows the overall architecture of our proposed model. The detailed description of the components is presented in the rest of this section.

4-1- Transliteration with Deep Sequence2sequence Model

A deep neural network is an artificial neural network with multiple layers between the input and output layers. Deep neural networks are powerful models with excellent performance in learning tasks [18]. Sequence2Sequence models are one of the well-known neural network-based models that are used for processing sequential data. In this model, a sequence is given to an autoencoder/decoder as an input and the system returns a sequence as an output. Both encoder and decoder parts of the model consist of Long Short Term Memory (LSTM) units, such that each unit supports one term in the input/out text.
The structure of a sequence2sequence model for machine translation is represented in Figure 2. The

* Corresponding Author

sequence2sequnce model in transliteration is similar to the translation model, but the units are characters instead of the words.

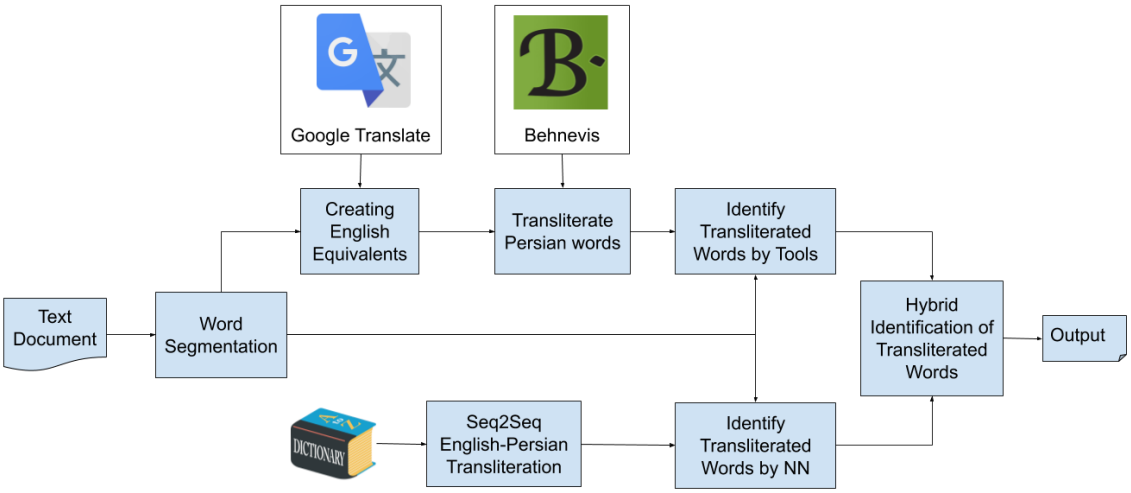


Fig. 1 Overall architecture of the proposed model

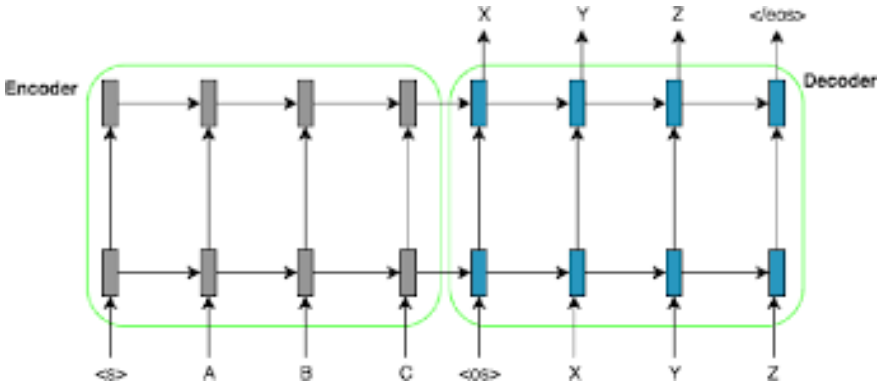


Fig. 2 A sample architecture for sequnce2sequence neural transliteration [9]

This method has been widely used in different natural language processing tasks, such as machine translation [3,1], question answering [11,15], dialog systems [23], and speech recognition [4]. In machine translation, an input sentence, as a sequence of words, is given to the system, and a sentence, again as a sequence of words, is generated in the output. The transliteration task has also the same behavior. It receives an input word, as a sequence of characters, and returns an output word, again as a sequence of characters [20].

To implement a sequence-to-sequence transliteration model, we used a 3-layer LSTM network as encoder, hidden, and decoder layers with Adam optimizer and learning rate 0.001. Each letter is represented by a 200-dimension embedding vector. The vectors are generated randomly and the representations are learned during the training phase. The implementation is done within the Tensorflow framework in Python. The model is trained on the English-Persian (EnPe) part of the transliteration shared task at Named Entities Workshop (NEWS) [24]. The dataset contains 14,758 training samples. Each sample consists of a pair of English and transliterated Persian words.

The trained model is then used to transliterate all words in the Persian input text into English. The transliteration is then compared to the English vocabulary. In case of any matching between the transliterated words and the English words in the dictionary, the word is detected as foreign word and the translation of the corresponding word in English-Persian bilingual dictionary is returns as the related word. The English vocabulary is selected based on the top 35k high frequent English terms from the Wall Street Journal Penn Treebank [12], and the Brown Corpus [19].

To be more precise, the proposed model captures the relationship between English and Persian words in a triangle as represented in Fig 3. The double line means that the main purpose of this model is to find the transliterated English word and its equivalent in Persian.

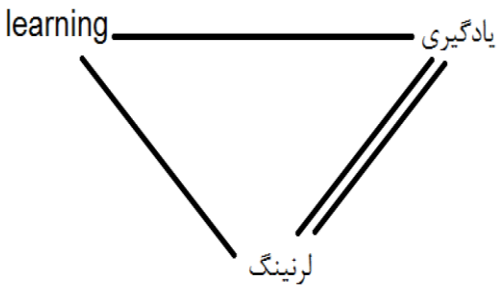


Fig. 3 a desired sample of the relationship between an original Englsih word (learning), its transliteration (لرنینگ), and the equivalent Persian word (یادگیری), which indicates the final goal of the research.: We want to find the transliterated English words in the Persian Language and its Persian equivalent. For example, "learning" is an English word and "لرنینگ" is its transliterated word in Persian. "یادگیری" is the Persian equivalent of "learning".

4-2- Tool-Based Transliteration

The tool-based approach tries to capture transliterated words by utilizing available toolkits that are compatible with this task. To this end, the tool-based approach is designed as follows such that the output of the first step is the input to the second step:

The first step is to find the English equivalent of the input words using *Google translate*. Google Translate is a free multilingual translation program developed by Google. It offers an API that helps developers build software applications. The tool is widely used in different researchers for question retrieval [17] as well as other text applications [22] and it is one of the best translators that can be used by programmers and researchers.

In this section, we translate each Persian word to English using the Google translate API. As it is known, Persian words are translated and foreign words are transliterated because these words come from foreign languages. Therefore, it is not possible to distinguish foreign words from original words; i.e., we cannot automatically find out if the output of the Google translate API is the translation of the Persian word to English, or it is the original English word that was transliterated to Persian. This question will be solved in the next steps.

The second step is to find the Persian equivalent of the English words using *Behnevis*. Behnevis is a website for transliterating English words into Persian. Most of the usage of this site is in the transliteration of Persian texts written in English. Here we use it to distinguish foreign words. In this way, if the word is given as an input to the Behnevis, the word is transliterated in Persian. For example, the word "بهینه", turns into "Optimum" in the first step and the second step becomes "اپتیموم". As another example, the foreign word "آمبولانس" becomes "Ambulance" in the first step and the second step again becomes "ambulance". As can be seen in these examples, the detection of transliterated words is based on the output word of Behnevis. In this way, if the word is an original Persian word, after translating it with Google Translate, the English translation will be returned. Google Translate passes the translated word to Behnevis, the word is spelled out and varies with the original input word. But if the word is foreign, Google Translate essentially transliterates the word to English and then Behnevis transliterates it back to Persian in the next step and returns the original word. Therefore, we need to compare the input word with the final word. If they are the same, the word is foreign and if they differ, it is a Persian word. Also, in this step, we can consider Google translate output as the equivalent of foreign words in Persian and English and reach the related word to Persian.

Although this method captures a large number of foreign words in texts, it still suffers from a shortcoming. Since some words have different pronunciations, they are transliterated to different forms when they come from foreign languages to Persian. Therefore, it is difficult to capture them with either of the proposed approaches. For example, by applying the above steps to the word "اورجینال" (the transliteration of the word "Original"), it becomes "اوریجینال" in which one character is added to it. We may also have such changes in consonants, such as "هندزفری" (the transliteration of the word "Hands-free"), which becomes "هندسفری" which replaces one letter. In such cases, although the transliteration is correct, the system cannot detect it, due to the difference between the final output and the input word. To solve this issue, we need to relax the exact matching between the terms and accept the matching between terms that differ in one character. To this aim, we use Levenshtein distance [10] algorithm to compare two strings and find the possibility of skipping one character when matching two words.

Levenshtein distance is used to calculate the difference between two strings. The output of the comparison of the two strings is the minimum number of changes needed to convert a string to another string. We have 3 actions, namely replacing a letter with another one, removing a letter, and adding a letter to the word. This algorithm works better than the Hamming algorithm because it calculates the distance between two words regardless of their lengths, whereas in the Hamming distance algorithm, the size of the 2 strings must be equal, and it only considers the replacement of characters [13].
Levenshtein distance uses dynamic programming, and its time complexity is O (m*n), where n and m are the lengths of two words.

5- Results

5-1- Dataset

To evaluate the performance of the proposed foreign word detection model, we provided a list of 979 words. The list consists of 491 Persian words and 488 foreign words. Persian words are those that have origin in the Persian language1. Foreign words are English words that are transliterated from English to Persian; i.e., all words are written with Persian alphabets.
It should be mentioned that in the input texts, there is no complete sentence and there is no need to word context information because the processing of each word is regardless of the previous and next word. Therefore, the proposed model can also be applied to single terms.

5-2- Evaluation Metrics

To evaluate our proposed model, we used 3 evaluation metrics, namely precision, recall, and F-measure. These metrics are calculated based on the confusion matrix of the results, as presented in Table 3.

Table 3: Confusion matrix

		Actual Values: English	Actual Values: Persian
Predicted Values	English	True Positive	False Positive
	Persian	False Negative	True Negative

[1]There a[illegible]abic words entered in Persian, however, since the characters are not identical with Persian, we can consider them

In the confusion matrix, true positive means the word is originally an English word and the system correctly identified as a transliterated word (English), true negative means the word is Persian and identified as Persian, false positive means the word is Persian but identified as English and false negative means the word is English but identified as Persian.
Considering the above descriptions, the metrics are calculated as follows:

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive + True negative}} \quad (1)$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive + False negative}} \quad (2)$$

$$\text{F-Measure} = \frac{\text{Precision . Recall}}{\text{Precision +Recall}} \quad (3)$$

5-3- Results of the Sequence2sequence Model

The first experiment was performed using sequence2sequence transliteration on the input text. The confusion matrix and the results of the model can be seen in Tables 3 and 4, respectively.

Table 4: Confusion matrix of output produced by the neural network method: There are 488 English words and 491 Persian words for testing.

	English	Persian
English	109 (22.33%)	14 (2.85%)
Persian	379 (77.67%)	477 (97.15%)

Table 5: Results of the foreign word prediction method using the neural network method

	English	Persian	Average
Precision	0.8862	0.5572	0.7217
Recall	0.2234	0.9715	0.5974
F-Measure	0.3568	0.7082	0.6537

As you can be seen in Table 3, the system has intention toward detecting words as Persian words rather than transliterated words. This results in high Precision in detecting English words and high recall in detecting Persian words. The overall f-measure of the system is 65.37%, which shows that the system has a correct prediction on about two-thirds of the words in the input list.
The neural network-based model is relatively fast in transliterating one word, but it needs to compare the transliterated word with all English vocabulary words

5-4- Results of the Tool-Based Model

A similar experiment has been done on the tool-based approach and the results are presented in Tables 5 and 6.

Table 6: Confusion matrix of output produced by the tool-based method: There are 488 English words and 491 Persian words for testing.

	English	Persian
English	275 (56.35%)	3 (0.61%)
Persian	213 (43.65%)	488 (99.39%)

Table 7: Results of the foreign word prediction method using the tool-based approach

	English	Persian	Average
Precision	0.9892	0.6961	0.8427
Recall	0.5635	0.9939	0.7787
F-Measure	0.7180	0.8188	0.8094

As can be seen in the tabulated results, although this method is slower than the neural network-based approach, its accuracy is very high and it detects about 80% of foreign words. Using this approach, we can see the following results for a sample English and Persian word respectively:

- "اسنک" ------> "snack" ------> "اسنک"
  prediction: English word
- "زبان" ------> "language" ------> "لنگویج"
  prediction: Persian word

The output of the tool-based approach shows that the second step of the model can clearly distinguish the words transliterated by Google from those that are translated.
Moreover, among not detected foreign words, we can see items with only 1 difference, so the recognition of foreign words even with minor differences can help to increase the accuracy of the system. The word "باینری" is an example of such words.

- "باینری" ------> "Binary" ------>" بینری"
  prediction: Persian word

Although these words are borrowed from English, due to differences between the source and target word, it cannot be detected as a foreign word with the proposed tool-based approach. Therefore, relaxing the exact match assumption and accepting one-character difference will help us to detect these words as well.
To this aim, in the next step of our experiment, we performed the tool-based approach while accepting 1 distance based on Levenshtein. The results of these experiments are presented in Tables 7 and 8.

Table 8: Confusion matrix of output produced by the tool-based method with accepting 1 Levenshtein distance: There are 488 English words and 491 Persian words for testing.

	English	Persian
English	364 (74.59%)	4 (0.81%)
Persian	124 (25.41%)	487 (99.19%)

Table 9: Results of the foreign word prediction method using the tool-based approach with accepting 1 Levenshtein distance

	English	Persian	Average
Precision	0.9891	0.7971	0.8931
Recall	0.7459	0.9919	0.8689
F-Measure	0.8505	0.8838	0.8808

5-5- Using the Joint Neural Network and Tool-Based Method

We can see in the results that the tool-based method outperforms the neural network-based method. But there is still room to improve the system by combining both models. To this end, we try to use the neural network-based transliteration besides the tool-based method. The results of the combined model that is applied to the same input text are presented in Tables 9 and 10.

Table 10: Confusion matrix of output produced by the joint method: There are 488 English words and 491 Persian words for testing.

	English	Persian
English	398 (81.55%)	18 (3.66%)
Persian	90 (18.45%)	473 (96.34%)

Table 11: Results of the foreign word prediction method using the joint approach

	English	Persian	Average
Precision	0.9567	0.8401	0.8984
Recall	0.8156	0.9633	0.8895
F-Measure	0.8805	0.8975	0.8939

As expected, comparing to the tool-based approach, the output improved and the F-measure of detecting English and Persian words increases 3% and 1%, respectively.
Given the fact that very few Persian words have been detected as foreign words; we hope that this method is working and we have greatly remedied the concern.

5-6- Error Analysis

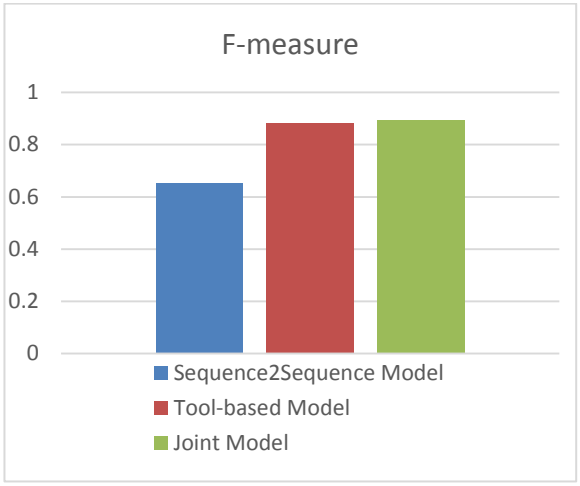
The error analysis of the words that cannot be recognized yet shows difficult items, such as, the word "دیلیت" (Delete). Although the proposed model achieved promising results in the task, there are some words that cannot be detected by our approach. To have a better understanding about our model. We had an error analysis and found the following issues the main reasons of the 11% error:

- For some words, the Google translate provides wrong translation instead of a correct transliteration. For example: the word "پرشین" is translated to "Jump" by Google translate. The NN output for this word is "perchin" which is wrong, too.

- The variation of letters in transliteration is more than one word and cannot be captured by our levenshtein distance approach. For example, the word "آداپتور" which is "adapter" in English is transliterated to "آدپتر" by Google translate, which is not completely wrong. Since in both languages the "a" vowel and the "ā" vowel are similar to each other, there can be more than one transliteration for a word and this makes the task difficult.

Overall, by using the combined model, our proposed system can correctly identify 89.39% of the words. For a better overall comparison, the average F-measure of all the models, including the hybrid model are presented in Figure 4.

Fig. 4 F-measures of Models respectively: Seq2Seq model, Tool-based model, Joint model



6- Conclusion and Future Work

Using the transliteration of foreign words instead of original Persian words becomes a common issue in recent years by the advent of Web2. This results in different problems when capturing the semantic meaning of a text. In this paper, we addressed this problem by proposing two different approaches and their combination. We showed that our proposed model, which benefits from a neural sequence2sequence model and a tool-based approach using Google translate and Behnevis APIs can detect foreign words and their corresponding Persian words with the F-measure of 89.39%.

As mentioned, although distributional approaches show great success in various tasks, their power comes from the data and the frequency of words in training corpora. The transliterated words, however, suffer from low frequency in training corpora and this is their main weakness to be used for the present work. In the future, we plan to use these approaches and try to adapt the models to be used for low-frequency words as well. We first try to discover the usage of ontology for the task [16] and then study more advanced contextualized techniques, such as BERT [5].

References

[1] P. Bahar, C. Brix, H. Ney. "Towards Two-Dimensional Sequence to Sequence Model in Neural Machine Translation." Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, p. 3009–3015.

[2] M. Bijankhan, J. Sheykhzadegan, M. Bahrani, M. Ghayoomi. "Lessons from building a Persian written corpus: Peykare", Language Resources and Evaluation, Vol. 45, No. 2, 2011, pp. 143-164.

[3] M.X. Chen, O. Firat., A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, N. Parmar, M. Schuster, Z. Chen, Y. Wu, M. Hughes. "The Best of Both Worlds: Combining Recent Advances in Neural Machine Translation." arXiv:1804.09849, 2018.

[4] C.C. Chiu, T.N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R.J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, M. Bacchiani, "State-of-the-Art Speech Recognition with Sequence-to-Sequence Models.", In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 4774-4778.

[5] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", In Proceedings of the NAACL-HLT Conference, Minneapolis, Minnesota, 2019, pp. 4171 - 4186.

[6] M. Ghayoomi, S. Momtazi, M. Bijankhan. "A study of corpus development for Persian." International Journal on Asian Language Processing, 2010.

[7] A. Hadifar, S. Momtazi, "The impact of corpus domain on word representation: a study on Persian word embeddings", Journal of Language Resources and Evaluation, Vol. 52, No. 4, 2018, pp. 997-1019.

[8] N. S. Kharusi and A. Salman, "The English Transliteration of Place Names in Oman". Journal of Academic and Applied Studies Vol. 1, No. 3, 2011, pp. 1–27.

[9] N.T. Le, F. Sadat, L. Menard, and D. Dinh, "Low-Resource Machine Transliteration Using Recurrent Neural Networks". ACM Trans. Asian Low-Resour. Lang. Inf. Process. Vol. 18, No. 2, 2019.

[10] *V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals". Soviet Physics Doklady. Vol. 10, No. 8, 1966, pp. 707–710.*

[11] B. Li, "A Question Answering System Using Encoder-Decoder Sequence-to-Sequence Recurrent Neural Networks.", Master Thesis, The Faculty of the Department of Computer Science, San José's State University, 2018.

[12] M. Marcus, B. Santorini, M.A. Marcinkiewicz. "Building a Large Annotated Corpus of English: The Penn Treebank", Journal of Computational Linguistics, Vo. 19, N. 2, 1993, pp. 313-330.

[13] F.P. Miller, A.F. Vandome, and J. McBrewster, Levenshtein Distance: Information Theory, Computer Science, String

(Computer Science), String Metric, Damerau? Levenshtein Distance, Spell Checker, Hamming Distance, 2009, Alpha Press.

[14] M. Mo'in, Mo'in Encyclopedic Dictionary, Amirkabir Publisher, 1972.

[15] A. Otsuka, K. Nishida, K. Bessho, H. Asano, and J. Tomita. "Query Expansion with Neural Question-to-Answer Translation for FAQ-based Question Answering". In Proceedings of the Web Conference (WWW), 2018. pp. 1063-1068.

[16] V. Reshadat, M.R.F. Derakhshi, "Studying of Semantic Similarity Methods in Ontology". Research Journal of Applied Sciences, Engineering and Technology, Vol. 4, No. 12, 2012, pp. 1815-1821.

[17] A. Rücklé, K. Swarnkar, I. Gurevych. "Improved Cross-Lingual Question Retrieval for Community Question Answering.", In Proceedings of the Web Conference (WWW), 2019, pp. 3179-3186.

[18] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview". Neural Networks. Vol. 61, 2015, 85–117.

[19] L. Schubert and M. Tong. "Extracting and evaluating general world knowledge from the Brown corpus". In Proceedings of the HLT-NAACL workshop on Text meaning, Association for Computational Linguistics, Vol. 9, 2003, pp. 7-13.

[20] I. Sutskever, O.Vinyals, Q.V. Le. "Sequence to Sequence Learning with Neural Networks." In Proceedings Neural Information Processing Systems (NIPS), 2014.

[21] Terminology Department, "A collection of terms approved by the Academy of Persian Language and Literature". Vol. 6, 2009, Academy of Persian Language and Literature. Tehran. (ISBN 978-964-7531-85-6)

[22] E.D. Vries, M. Schoonvelde, and G. Schumacher, "No Longer Lost in Translation: Evidence that Google Translate Works for Comparative Bag-of-Words Text Applications". Political Analysis, Vol. 26, No. 4, 2018, pp. 417-430.

[23] B. Wei, S. Lu, L. Mou, H. Zhou, P. Poupart, G. Li, Z. Jin, "Why Do Neural Dialog Systems Generate Short and Meaningless Replies? a Comparison between Dialog and Translation", In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. pp. 7290-7294.

[24] M. Zhang, H. Li, A. Kumaran, M. Liu, "Report of NEWS 2011 Machine Transliteration Shared Task". In Proceedings of the 5th international Joint Conference on Natural Language Processing (lJCNLP), 2011.

Saeedeh Momtazi is currently an assistant professor at the Amirkabir University of Technology, Iran. She completed her BSc and MSc education at the Sharif University of Technology, Iran. She received a Ph.D. degree in Artificial Intelligence from Saarland University, Germany. As part of her Ph.D., she was a visiting researcher at the Center of Language and Speech Processing at Johns Hopkins University, US. After finishing the Ph.D., she worked at the Hasso-Plattner Institute (HPI) at Potsdam University, Germany and the German Institute for International Educational Research (DIPF), Germany as a postdoctoral researcher. Natural language processing is her main research focus. She has worked in this area of research for more than 15 years.

Ali Hoseinmardy received the B.S. degree in Computer Engineering from Amirkabir University of Technology, Tehran, Iran in 2018. Currently he Is a MBA graduate student in Sharif University of Technology, Tehran, Iran. His research interests Natural Language Processing, Data mining, Organizational Leadership and Gamification.