IJICTR International Journal of Information & Communication Technology Research

ITRC

Volume 4- Number 5- December 2012

Persian to English Personal Name Transliteration Based On the Persian Web Contents

Zohre Haghollahi
Electronic and Computer engineering department
Yazd University
Yazd, Iran
zohre.haghollahi@stu.yazduni.ac.ir

Ali Mohammad Zareh Bidoki
Electronic and Computer engineering department
Yazd University
Yazd, Iran
alizareh@yazduni.ac.ir

Alireza Yari
Iran Telecommunication Research Center
Tehran, Iran
a_yari@itrc.ac.ir

Received: February 12, 2012- Accepted: November 3, 2012

Abstract—**Personal names are out of dictionary words which are usually primary keys in the web queries. Converting a personal name from source language to target language is transliteration. In this paper, we propose a novel algorithm for transliterating a Persian personal name to an English name. This method consists of two stages. At first, in the offline stage, a graph is made by processing Persian web contents. In this graph, names are related based on their neighboring in the web pages. Afterwards, in online stage, each name is transliterated using this graph and name frequencies. Experimental results show that the algorithm is effective in personal name transliteration.**

Keywords-*Transliteration; Cross Language Information Retrieval; Machine Translation;Personal names;Graph*

I. INTRODUCTION

Machine Translation (MT) is one of the fundamental parts of any multilingual applications such as Cross Language Information Retrieval (CLIR), Cross Language Question Answering (CLQA) and etc. The basic solution for MT is bilingual dictionary. But a large number of words are out of dictionary. These terms are typically names like people, organization, places and products. To convert an out of dictionary word to another language, transliteration is done. Transliteration (Translation-by-sound [1]) is the process of transforming the script of a word from a source language to a target language, based on how it is pronounced in each language [2]. As mentioned earlier, personal names are a special fragment of the out of dictionary words which are frequent in web queries. According to [3], 2~4% daily web queries are in the form of exact personal names. The count is even higher if we consider all queries that contain personal names which is about 30% [4]. This also can be doubled when some people enter a query in a search engine and do not get satisfying results, so they switch to another search engine. There is another reason which makes the personal names important. Many people put their names as a query and estimate the performance of search engine. Thus, personal names play a significant role in the web queries.

As we know, many people design their home pages not in their native languages. Therefore, recognizing [3,5] and thereupon transliterating a

personal name to another language can be an effective way to improve search results, because users can access people's homepages in different languages.

In this paper, we concentrate on personal names and present a novel algorithm for Persian to English personal name transliteration. Persian (Farsi) is an Indo-European language written in Arabic script from right to left, but with an extended alphabet and different pronunciation from Arabic [6]. Here, a source query is a Persian personal name that is composed of a first name and a last name. It is shown as $Q_{in} = t_1t_2 \dots t_i$, $2<i<n$ where n is the number of terms in Q. Q_{out} is also English form of Q_{in}. For example, Q_{in}="حمید زارع زاده" is converted to Q_{out}="Hamid Zareh Zadeh".

An alternative and straight-forward solution for the problem of translating the web queries is using dictionary. Given a query, we can check whether the contained terms are in some predefined terms in dictionaries. But some name terms such as personal names, places, organizations and etc do not exist in dictionaries. To translate such a word, we should apply transliteration rules. Since personal names are frequent in the web queries, therefore we focus on Persian to English personal names transliteration.

There are two challenges in Persian to English transliteration which makes it difficult. First, In Persian script short vowels aren't displayed and only long vowels are shown. Therefore to convert a Persian word to an English word, we should know short vowels. For example "حمید" is written "Hamid" in English but the short vowel "a" doesn't appear in the English form. Second, the evaluation process of transliteration is not straightforward, because transliteration allows multiple variants of a source term to be valid, based on the opinions of different human transliteration [7]. A Persian name "محمود" is transliterated to "Mahmood", "Mahmoud" and "Mahmud".

To solve the first problem, we proposed a novel algorithm to transliterate Persian personal names to English. In this algorithm, many Persian web pages are selected. By processing these pages, lots of words are extracted and a neighboring graph is made. Each node is assigned a root by ignoring some vowels of the names. Creating this graph is performed in an offline stage. In the online stage, transliteration is done using this graph. For the second problem, we also prepare a Persian to English name corpus which contains various modes of a name transliteration. In the following section, the algorithm is described in more details.

The rest of the paper is organized as follows. In Section Π, we present some previous work related to transliteration. We refine our problem description in Section Ш. Section IV describes our proposed method. We study the performance of our method in Section V. Finally, we conclude our work in Section VI.

II. Related Work

Transliteration is a sub field of computational linguistic and study on this field has begun approximately since ten years ago. According to [7], Transliteration method is divided into two groups: generative transliteration and transliteration extraction.

Generative transliteration is also classified to four groups which are phonetic-based methods [8,9], spelling-based methods [10], hybrid [11], and combined methods [12,13]. Most early studies on transliteration were performed in a phonetic-based (or phoneme-based) framework. The intuition behind this method is that phonetic representation is an intermediate form between source ad target languages. But this approach relies on bilingual pronunciation resources which are not available for all languages [7]. Unlike phonetic-based approaches, spelling-based methods (or grapheme-based) transform a groups of characters in the source language to a group of characters in target language. Hence, there is a *direct* mapping between source and target languages. Since spelling-based techniques in comparison to grapheme-based approach reduce the number of steps involved in transliteration, they can remove some potential sources of errors in overall process. The next approach is Hybrid method which tries to combine the two previous techniques [14,15]. In some experiments, hybrid methods have shown significant improvements in comparison to previous methods [7]. In combined methods some independent system outputs are combined and a number of candidate answers will be obtained. In [12], for English to Persian transliteration and Back-transliteration, several spelling-based systems were aggregated to one system using combination method based on a mixture of a Naïve-Bayes classifier and a majority of voting schemes. It should be noted that this technique has been recently used for machine transliteration.

The second approach of machine transliteration is transliteration extraction. In this method, transliteration pairs from multilingual resources like parallel or comparable corpora and web are extracted. These pairs are exploited based on web co-occurrences [16,17], phonetic similarity [18] or the other machine learning techniques such as active learning[18] and adaptive learning [19].The advantage of this approach is that the transliteration pairs can be used for training generative systems, which then allows previously unseen terms to be proceeded [7].

We can see that the first approach for machine transliteration is done on the grapheme or phoneme of a word. But short vowels do not appear in Persian script; therefore these methods are not suitable for this language. The second approach is also complex and extracting word pairs are time consuming. Hence, we propose a method which is simple and also solves the challenges of Persian to English transliteration.

III. Proposed Algorithm

In this paper, we propose an approach for personal name transliteration based on the personal names graph and names frequencies. Figure 1 provides an overview of our approach, which works under two stages: the offline stage and the online stage. In the

offline stage, we construct a graph including the personal names. Given many Persian web pages, we filter a number of words to create the graph. The online stage is for transliteration, in which we can have the transliteration of the source query (Q_{in}) to the target query (Q_{out}). Our approach is expected to exploit possible candidates for transliteration effectively in the online stage. In section A, a number of concepts are explained. Then in section B, the offline stage is described and the online stage is clarified in section C.

A. Concept definition

Our goal is to create a graph in the offline stage and use it to transliterate a personal name. Before coming to the details of the methods, we define several concepts for the convenience of description:

Bilingual Dictionary Filter (D-Fil): This filter is an English to Persian dictionary. Each word which is not in this dictionary passes D-Fil. Because names that are not in dictionary, go to the next step. For example the word "Hamid" is not in dictionary and it goes to the next step but the word "book" exists in dictionary and is ignored.

Not English Filter (NE_Fil): This filter is for recognizing the words which have English letters. For example, the word "Ali" passes the filter to the next step but "فناوری" is removed.

Hidden Vowel (H-Vowel): We define {a, u, o, e} as hidden vowel letters because these letters may not appear in Persian script.

root (r): root of a word is obtained by removing H-Vowels from middle letters, not the first and last letters. For example the words "mobina" and "mubina" are in root "mbina" or the words "davoud", "davood", and "dvd" are in root "dvd".

Frequency (Freq): Freq(w) is the number of w met in Persian web pages.

Distance (D): D(x,y) is distance between the words x and y in Q_{in}. For instance, If Q_{in} = " حمید زارع زاده", D("حمید","زارع") = 1 and D("حمید","زاده") = 2.

B. Stage one (Offline Stage)

In this section, we propose a method to create a graph in the offline stage. First, we explain about the reason for choosing this method. Then the algorithm is clearly described.

The principal idea of our method is based on two notions. First, short vowels do not appear in the Persian script and second, first names and last names are close together in a text. For the first notion, we introduce a new concept as "root". As mentioned previously, root of a word is obtained by removing middle H-Vowel letters. Therefore, the words which are the same in their consonant letters are grouped together. This can be useful to select the candidates of transliterated word. This notion is described in the next section. The other notion was that first names and last names are in a neighborhood. Hence, we consider the neighbor of a word. These relations can be depicted by a graph and is used to choose best result among a list of candidates. It will also be described in the following.

A graph is G = (V,E) which V is the Vertices of the graph and E refers to Edges. Vertices are Finglish words[1] and Edges are relationships between them. To create the graph, all the words of Persian web pages are extracted. These words should pass two filters. First the letters of word is checked to see if it is

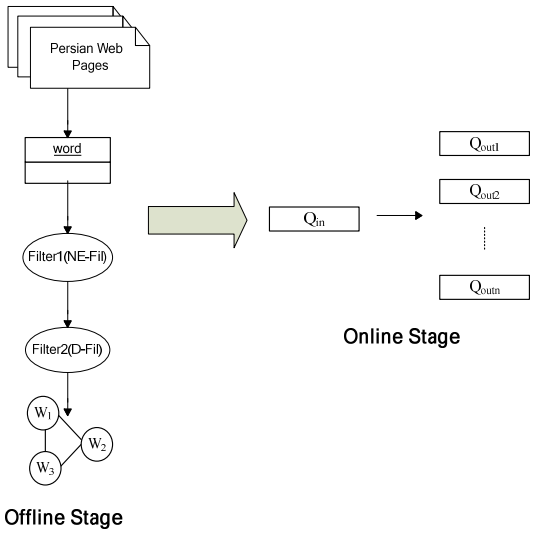

Figure 1. Two stages of the proposed transliteration system

English. If the word contains English letters, it goes to the next filter. The reason of using this filter is that we want to create a graph containing English forms of Persian personal names such as "Ali" or "Moradi" and find their relationships. Afterwards, we check its existence in bilingual dictionary. If the word exists in dictionary, it will be ignored for the next steps, otherwise; it passes to create the graph. This filter is for removing the words that are not personal names definitely because personal names do not exist in dictionaries. Actually, we can't assert that all the nodes are personal names but we remove many words which are not certainly of this kind. The extracted words from previous steps form the nodes of graph and they are labeled by their roots. The links between the nodes are based on the neighboring in Persian web pages. If two words are seen close together, a link is formed between them. If these nodes are met again, the link between them gets stronger or its weight increases. Let's define an example. As shown in figure 2, we have 7 nodes which are labeled by their roots. The links between them are formed based on the neighboring. For instance, if the initial value for the weight be 0.05 and the weights increases 0.05 for every seen pairs, "ghasem" and "karami" are met 4 times near each other. Because the weight of the link between these nodes is 0.2, but "nujood" and the other nodes aren't seen close together and there is no link between them.

While processing Persian web pages to get the words, we should also consider the frequency of each word in the whole web pages.

In this stage, two factors are obtained from Persian web pages: neighboring of the words and words

[1] Finglish word is a Persian word in English script. For example "ali" is a Finglish word for "علی".

frequency. These two factors are considered baseline approaches in online stage.

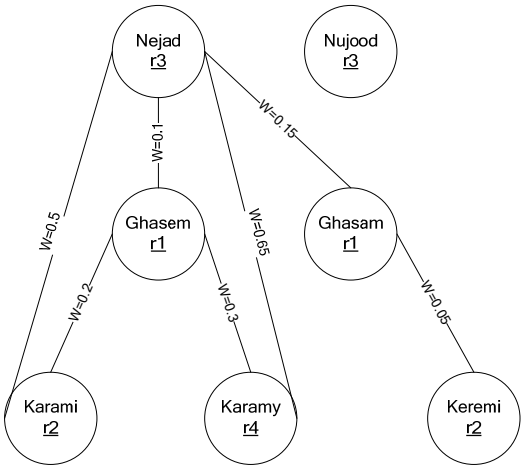

Figure 2. An example of neighbouring graph

C. Stage two (Online Stage)

As mentioned, one of the problems of Persian to English transliteration, which makes it difficult, is missing short vowels in Persian script. To solve this problem, in the offline stage, we created a graph in which nodes are the Finglish words and links are relations between names. Also a root is assigned to each word. The main goal of Online Stage is transliterating a Persian personal name to English by using a mapping function and offline stage outputs.

To transliterate a Persian personal name to English in the online stage, a mapping function transforms letters of a Persian name to English script; on the other hand, this function converts the word to its appropriate roots Figure 3 shows the mapping function. For example, the word "زهره" is converted to "zhrh" or the word "وحید" is also transformed to "vhid", "ohid", "vhyd" and "ohyd". In this example, the term is transformed to more than one root; therefore a number of candidates are considered for it. Then we put the names related to each root in the R set. In the former example, the root members of "zhrh" are R={zohreh, zahereh, zuhreh}. Now we should select the best result through these candidates. We apply three methods to get the best result. These methods are based on three factors: first, frequency of the candidates in the Persian web pages. second, relationship between each candidate and the candidates of the other terms in Q_{in} and third hybrid approach of the two factors.

The first factor is the frequency of the candidate results. Frequency of a word is an important factor to consider. Because if a word is frequent in web, we conclude that it can be a right selection for our result. So the probability of candidate in R to be the transliteration of term t_i is calculated as:

$$p_1(R_{ij} \mid t_i) = \frac{\log(freq(R_{ij}))}{\sum_{j=1}^{m} \log(freq(R_{ij}))} \qquad (1)$$

The desired candidate is R_{ij} that is jth member of root i and t_i is ith term in Q_{in}. m is also the number of candidates in root i. Because frequency is a large number, we use logarithm.

gh <-- ق		r <-- ر		a,e,o <-- ا	
k <-- ک		z <-- ز		b <-- ب	
g <-- گ		s <-- س		p <-- پ	
l <-- ل		sh <-- ش		t <-- ت	
m <-- م		s <-- ص		s <-- ث	
n <-- ن		z <-- ض		g <-- ج	
v,o <-- و		t <-- ط		ch <-- چ	
h <-- ه		z <-- ظ		h <-- ح	
i,y <-- ی		a,e <-- ع		kh <-- خ	
		gh <-- غ		d <-- د	
		f <-- ف		z <-- ذ	

Figure 3. Persian to English mapping function

The other factor is the words relationship in the graph. Because first names and last names are close together in a text, so if the relationship between two candidate words is stronger or they are seen close together, we conclude that they can be a right selection for the result. So the weight between desired candidate and the other terms candidates in Q_{in} is considered. The distance between the terms in Q_{in} is also important, because the close terms have a stronger relationship with each other. This is shown in formula 2:

$$p_2(R_{ij} \mid t_i) = \frac{\sum_{\substack{s=1 \\ s\sim=i}}^{n} \sum_{q=1}^{d} W(R_{ij}, R_{sq}) / D(R_{ij}, R_{sq})}{\frac{1}{2} \sum_{j=1}^{m} \sum_{\substack{s=1 \\ s\sim=i}}^{n} \sum_{q=1}^{d} W(R_{ij}, R_{sq}) / D(R_{ij}, R_{sq})} \qquad (2)$$

In which n is the number of terms in Q_{in}, d is the number of members in each root s, $W(R_{ij},R_{sq})$ is the weight between R_{ij} and R_{sq}, $D(R_{ij},R_{sq})$ is the distance between R_{ij} and R_{sq} in Q_{in} and m is the number of candidates in root i. So the probability is calculated by considering all relationships between R_{ij} and members of other term's roots. On the other hands, the aggregate of weights between R_{ij} and members of other term's roots is divided to total sum of all relationships between every candidate to get the probability.

Another method to select the best result is hybrid of two previous baseline algorithms. Therefore, the frequency of the candidates and their relationship in graph is considered. The probability is also calculated as:

$$p_3(R_{ij} \mid t_i) = \frac{\log(freq(R_{ij})) + \sum_{\substack{s=1 \\ s\sim=i}}^{n} \sum_{q=1}^{d} W(R_{ij}, R_{sq}) / D(R_{ij}, R_{sq})}{\sum_{j=1}^{m} \log(freq(R_{ij})) + \frac{1}{2} \sum_{j=1}^{m} \sum_{\substack{s=1 \\ s\sim=i}}^{n} \sum_{q=1}^{d} \frac{W(R_{ij}, R_{sq})}{D(R_{ij}, R_{sq})}} \qquad (3)$$

Which is the hybrid of two previous formulas.

Algorithm 1 provides an overview of the hybrid approach. In this algorithm, first the source query is converted to English candidate queries by a mapping function and afterward the probabilities are selected according to the frequency and relationship in graph. Then each term in Q_{out} is the candidate which has maximum probability.

Algorithm 1: Hybrid Approach of Online Stage

FUNCTION **TRANSLITERATION**

INPUT :

i. Graph;
ii. Q_{in} ($t_1t_2…t_n$); //Input Query

OUTPUT :

i. Q_{out} ($o_1o_2…o_n$); // Output Query

FOR i=1 : n
$r_{i1}r_{i2}…r_{im}$=root of t_i by mapping function;
R_i = members of r_{ij}, 1<j<m;
END FOR

FOR i=1: n

$$total_i = \sum_{j=1}^{m} \log(freq(R_{ij})) + 1/2 \sum_{j=1}^{m} \sum_{\substack{s=1 \\ s\sim=i}}^{n} \sum_{q=1}^{d} \frac{W(R_{ij},R_{sq})}{D(R_{ij},R_{sq})};$$

$$p(R_{ij} \mid t_i) = \frac{\log(freq(R_{ij})) + \sum_{\substack{s=1 \\ s\sim=i}}^{n} \sum_{q=1}^{d} W(R_{ij},R_{sq})/D(R_{ij},R_{sq})}{total_i};$$

$$o_i = \max_j(p(R_{ij} \mid t_i)), 1 < i < n;$$

END FOR

TABLE I. WEIGHTS BETWEEN CANDIDATES(FREQUENCY)

	Karami (700)	*Keremi (16)*	*Karamy (430)*	*Nejad (850)*	*Nujood (30)*
Ghasem (1500)	0.2	0	0.3	0.1	0
Ghasam (440)	0	0.05	0	0.15	0

For further understanding of the subject, let us give an example. Suppose Q_{in} = "قاسم کرمی نژاد".The first step is assigning root (or roots) to each term by mapping function.

t_1=قاسم → r_{11}=ghsm

t_2 = کرمی → r_{21} = krmi , r_{22} = krmy

t_3 = نژاد → r_{31} = njd

Then the members of each root, obtained from offline stage, are set in R. Figure 2 presents the query terms and their relationship.

r_{11} (ghsm) = {ghasem,ghasam} = R_1
r_{21} (krmi) = {karami, keremi}
r_{22} (krmy) = {karamy} } = R_2
r_{31} (njd) = {nejad, nujood} = R_3

We want to select the best result for "ghsm" which the candidates are {ghasem(R_{11}), ghasam(R_{12})}. Table I shows the weights between the candidates.

$$total_1 = \log(1500) + \log(440) + (\frac{0.2}{1} + \frac{0}{1} + \frac{0.3}{1} + \frac{0.1}{2} + \frac{0}{1}) =$$
$$3.17 + 2.64 + 0.05 = 5.86$$
$$P(R_{11} \mid t_1) = \frac{\log(1500) + 0.2/1 + 0/1 + 0.3/1 + 0.1/2 + 0/2}{5.86} =$$
$$\frac{3.72}{5.86} = 0.63$$
$$P(R_{12} \mid t_1) = \frac{\log(440) + (0/1 + 0.05/1 + 0/1 + 0.15/2 + 0/2)}{5.86} =$$
$$\frac{2.64 + 0.125}{5.86} = 0.47$$

As shown, "ghasem" is selected as the best result because the probability of "ghasem"(0.63) was greater than the probability of "ghasam"(0.47). The other term's Transliterations are also calculated as above.

D. Complexity of the proposed algorithm

As discussed, the proposed algorithm has two stages: the offline and the online stage. The complexity of the offline stage is not important. Because it works independent of the user's query.

The complexity of the online stage is calculated based on the Algorithm 1. Here, n is the number of terms in Q_{in}, m is the number of input query roots and d is also the number of members in each root. The first loop concludes a mapping function to obtain roots and getting members of each root. Obtaining roots is based on a mapping function. So the complexity is O(1). We also get the members of each root in offline stage, therefore the complexity of getting a root member is O(1). This loop iterates n times, so the complexity is O(n*1) = O(n). In the second loop, the complexity of getting $total_i$ is O(m+m*n*d) = O(m*n*d) and the complexity of getting $p(R_{ij}|t_i)$ is O(n*d+m*n*d)=O(m*n*d). Obtaining each output query term by getting the maximum of probabilities is O(n).

The complexity of Algorithm 1 or online stage is obtained by considering the above calculations. So we sum up these complexities: O(n) + O(m*n*d) + O(m*n*d) + O(n) = O(m*n*d). Therefore the complexity of online stage is O(m*n*d).

IV. EXPERIMENTS

A. Data prepration

In this section, some experiments are done to test our system. To do so, we need a number of Persian web pages and also a corpus to test our system.

We use 1,360,000 Persian web pages indexed by parsijoo search engine [20]. These pages are obtained from SID [21], IRANDOC [22] and CIVILICA [23]. We selected these sites because they are a resource for scientific papers and contain many Finglish names.

Persian to English name pairs transliteration corpus is not available. Hence we prepare it by hand. The corpus consists of 600 Persian first names and last names and all of the possible English transliteration for each name. Figure 4 shows a part of our corpus. As discussed in section II, a number of names have more

than one transliteration and it is shown here. So the second problem of Persian to English transliteration is solved by this corpus, because it has the entire transliteration modes of a Persian name. Suppose the queries are in the form of first name and last name, 400 combinations of first names and last names are

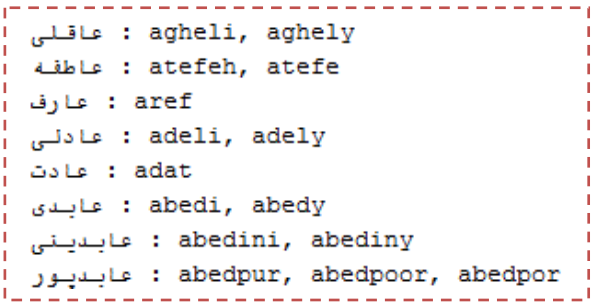

Figure 4. Persian to English name corpus

It should be noted that for the D-Fil in offline stage an English to Persian bilingual dictionary, which consists of 60700 English words, is used.

B. Evaluation Metric

We employ the standard measure to evaluate the performance of name transliteration in web queries. i.e. word accuracy (transliteration accuracy or precision) [10]. According to formula 4, word accuracy is the proportion of names which were correctly transliterated among all names.

$$WordAccuracy = \frac{\#CorrectTransliteration}{\#TestNames} \qquad (4)$$

Top-n word accuracy indicates the proportion of words in the test set in which the correct transliteration was returned within the first n candidate answers.

C. Results and analysis

We study our proposed method and present the experimental results as well as the analysis.
In online stage, the best results are selected based on three approaches: words frequency method (Freq), graph links Method (Link) and a hybrid approach of frequency and graph links (Hybrid) as discussed in previous sections. Table II presents the comparison results among three approaches. As mentioned before, the evaluation metric of results is Top-n word accuracy. For instance, Top-1 signifies the proportion of words in the test set for which the correct transliteration was the first candidate answer returned by the system.

TABLE II. WORD ACCURACY (%) OF THREE APPROACHES IN ONLINE STAGE

	Top-1	*Top-3*	*Top-5*
Freq	70	83	**87**
Link	62	73	81
Hybrid	**74**	**85**	**87**

From table II, we can see that hybrid method outperforms the two others especially in top-1 word accuracy. This is particularly important in the systems which expect the best result. The comparison of Freq and Link indicates that Freq has better outcome than Link in all experiments. It is because a number of first names and last names in the test set may not have links in graph so word accuracy in link is less than Freq approach.

As discussed in introduction, the only Persian to English transliteration system was developed in [12]. This method is named as spelling. We compare our hybrid approach with the spelling method. The results are shown in table III.

TABLE III. COMPARISON BETWEEN WORD ACCURACY (%) OF HBRID AND SPELLING APPROACHES

	Top-1	*Top-3*	*Top-5*
Hybrid	**74**	**85**	**87**
Spelling	69	80	84

V. As shown in the above table, our approach is better than the spelling method in all Top-n Word Accuracies. The main reason is that we consider the words of web which is very huge and complete. This corpus contains a lot of Finglish words, so that the precision of our algorithm is increased. The other reason is that in our approach the relations between the first names and last names are considered.

VI. CONCLUSION AND FUTURE WORK

In this paper, we studied the problem of personal name transliteration in web queries. To do so, we propose an algorithm which contains two stages. In the first stage, a graph is made by processing Persian web contents and then a name is transliterated in online stage. This method is compared by two baseline methods and experiments show that our proposed method outperforms the two others particularly in Top-1 word accuracy.

To construct the graph in offline stage, we hope to exploit more information about the names and their relationship in web pages. Besides that, we will also try to increase the number of web pages to get better results.

ACKNOWLEDGMENT

This work was supported by Iran Telecommunication Research Center.

REFERENCES

[1] J.S Kuo and H. Li, "Multi-View Co-Training of Transliteration Model," The Third International Joint," In Conference on Natural Language Processing (IJCNLP-08), pp. 373-380 , Hyderabad, India, January 2008.

[2] L.Th. Hoang Diem and A.A Ti, "Lexical word similarity for re-ranking in Vietnamese-English named entity back-transliteration, " In 2011 International Conference on Asian Language Processing (IALP), pp. 197-200, Penang, November 2011.

[3] D. Shen, T. Walker, Z. Zheng, Q. Yang and Y. Li, "Personal Name Classification in Web Queries," In First ACM International Conference on Web Search and Data Mining (WSDM '08),stanford, February 2008.

[4] J. Artiles, J. Gonzalo, and F. Verdejo. "A testbed for people searching strategies in the WWW," In SIGIR '05: Proceedings

of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 569-570, New York, NY, USA, 2005.

[5] Z. Haghollahi, A.M. Zareh Bidoki and A. Yari, "Personal names recognition in query using search engine," In 17's Conference on Computer Society of Iran, pp. 144-149, Tehran, Iran, March 2012.(in persian)

[6] S. Karimi, F. Scholar and A. Turpin, "Collapsed Consonant and Vowel Models: New Approaches for English-Persian Transliteration and Back-Transliteration," In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pp. 648-655, Prague, Czech Republic, June 2007.

[7] S. Karimi, F. Scholar and A. Turpin, "Machine Transliteration Survey," ACM Computing Surveys, Vol. 43, No. 3, Article 17, pp. 1-46, April 2011.

[8] W. Gao, K.F. Wong and W. Lam. "Phoneme-based transliteration of foreign names for OOV problem," In Proceedings of the 1st International Joint Conference on Natural Language Processing(IJCNLP'04), Lecture Notes in Computer Science, Vol. 3248, Springer, pp. 110-119, , Berlin, 2004.

[9] P. Virga and S. Khudanpur, "Transliteration of proper names in cross-lingual information retrieval," In Proceedings of the ACL Workshop on Multilingual and Mixed-Language Named Entity Recognition, Vol. 15, pp. 57-64, Stroudsburg, PA, USA, 2003.

[10] S. Karimi, A. Turpin and F. Scholer, "English to Persian transliteration," In String Processing and Information Retrieval. Lecture Notes in Computer Science, Vol. 4209, Springer, pp. 255-266, Berlin, 2006.

[11] J.H. Oh, K.S. Choi, and H. Isahara, "A machine transliteration model based on correspondence between graphemes and phonemes," In ACM Transactions on Asian Language Information Processing (TALIP). Vol. 5, Issue 3, pp. 185-208, New York, NY, USA, September 2006.

[12] Karimi, S. "Machine transliteration of proper names between English and Persian," Ph.D. dissertation, RMIT University, Melbourne, 2008.

[13] J.H. Oh and H. Isahara, "Machine transliteration using multiple transliteration engines and hypothesis re-ranking," In Proceedings of the 11th Machine Translation Summit. pp. 353-360, Copenhagen, Denmark, September 2007.

[14] Y. Al-Onaizan and K. Knight, "Machine transliteration of names in Arabic text," In Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages. pp. 1-13, Stroudsburg, PA, USA, 2002.

[15] S. Bilac, and H. Tanaka, "Direct combination of spelling and pronunciation information for robust backtransliteration," In Proceedings of the Conferences on Computational Linguistics and Intelligent Text Processing, pp. 413-424, 2005.

[16] W. Lam, R. Huang, and P.S. Cheung, "Learning phonetic similarity for matching named entity translations and mining new translations," In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, pp. 289-296. 2004.

[17] J.H. Oh, and K.S. Choi, "An ensemble of transliteration models for information retrieval," Information Processing and Management: an International Journal. Vol. 42, Issue 4, pp. 980-1002, Tarrytown, NY, USA, July 2006.

[18] J.-S. Kuo, H. Li, and Y.K. Yang, "Active learning for constructing transliteration lexicons from the Web," In Journal of the American Society for Information Science and Technology, Vol. 59, Issue 1, pp. 126-135, January 2008.

[19] H. Li, J.S. Kuo, J. Su, and C.L. Lin, "Mining live transliterations using incremental learning algorithms," In Int. J. Comput. Process. Oriental Lang. Vol. 21, No. 2, pp. 183-203, 2008.

[20] Parsijoo Search Engine. http://www.parsijoo.ir

[21] Scientific Information Database. http://www.sid.ir

[22] Iranian Research Institute for Information Science and Technology-IRANDOC. http://www.irandoc.ac.ir

[23] Encyclopedia of civil engineering-CIVILICA. http://www.civilica.com

Zohre Haghollahi got her B.Sc. and M.Sc. degrees in Computer Engineering from Yazd University in Iran. Her research interests are Search Engine, Cross Language Information Retrieval and Translation and Transliteration Systems. Currently she is working on translation service of search engine.

Ali Mohammad Zareh Bidoki got his B.Sc. degree in Computer Engineering from Isfahan University of Technology in Iran. He also received his M.Sc. and Ph.D. degrees from the University of Tehran in Iran. His research interests include web information retrieval, search engines and data mining. He has published more than 25 papers in international journals and conferences. Currently, he is an assistant professor in Yazd University, Iran.

AliReza Yari received his B.Sc. degree in Control System Engineering in 1993 from the University of Tehran, Iran, and his M.Sc. and Ph.D. degrees in systems engineering in 2000 from Kitami Institute of Technology, Japan. He is currently doing research in Information Technology Research Faculty of Iran Telecom Research Center (ITRC). His research interests include web information retrieval and language specific search engine, e.g. Persian search engine. He is also working on application of natural language processing (NLP) for improvement of search engines.

This Page intentionally left blank.