English-Korean Machine Transliteration by Combining Statistical Model and Web Search

Hyun-Je Song and Seong-Bae Park

***Abstract*—Machine transliteration is an automatic method for translating source language words into phonetically equivalent target language ones. Many previous methods were devoted to translating the word that only traces phonological phenomena of the source language and the resulting showed good performance. However, there are a lot of names originated from not only the source language but also non-source languages. The existing methods fail in showing high accuracy when the names comes from the non-source language since they focus on names in source language. To deal with this problem, this paper describes a hybrid method which combines statistical model and web search for improving machine transliteration performance. The proposed method constructs a base system that stands on a statistical model to produce candidates, then expands candidates from web documents. With these candidates, it finds the most appropriate answer without any external resources. The experimental results present that the proposed method achieves higher performance than statistical model and web search respectively.**

***Index Terms*—Machine transliteration, Statistical model, Web search**

I. Introduction

MACHINE transliteration is the conversion of a given name in source language to a name in target language such that the target language name is phonemically equivalent to the source name [1], [2]. There has been growing interest in the use of machine transliteration since it is a tool to support various applications such as cross-language information retrieval and machine translation [3]. This paper presents the automatic English-Korean forward transliteration that the source language is English and the target language is Korean1[1]. For example, given a English name ‘Smith’, it is transliterated into a Korean name ‘스미스’, and it translates the national name ‘Brazil’ as ‘브라질’.

In general, machine transliteration does not use the context information, unlike several natural language processing problems such as machine translation and sentence parsing, because of the nature of machine transliteration. Therefore, it is difficult to translate the source name into only one target name directly without the context information.

To tackle the problem simply, in this paper, machine transliteration is divided into two steps: candidate generation and answer search. In candidate generation, given the name in source language, a generate model generates candidates which have the possibility of being an answer. The answer search step finds the most appropriate answer from the candidates which are generated in previous step. This step is regarded as a ranking step since search models use ranking functions for calculating the possibility that each candidates are to being an answer.

There have been proposed various approaches for machine transliteration. They were devoted to translating the word that only traces phonological phenomena of the source language. In this situation, they showed good performance. However, there are a lot of names originated from not only the source language but also non-source languages. While they are written with the source language, its phonetic does not conform to the pronunciation of source language. Let consider an name ‘Naples’ for example, which is a city in Italy. It should be transliterated as the Korean name, ‘나폴리’. The previous methods based on statistical model only transliterate ‘Naples’ as ‘네이플스’, ‘나플레스’, and so on, not ‘나폴리’, since they reflect the most prevalent transliterations among the bilingual corpus, that comply with the English to Korean transliteration notation. The rule-based approaches with conversion rules corresponding to the origin do not also hold in this case because there is no way to know the origin of name exactly.

As a solution of this problem, this paper proposes a model which combines a statistical model and a web search to translate names that originated from not only the source language but also non-source languages. The proposed method first generates candidates using statistical model. Second, words to be the answer are added through searching web documents. By combining the statistical model and the web search, candidates of the name derived from non-source languages are generated. In order to search an correct answer among candidates, the proposed model finally ranks candidates with a machine learning based ranking function. To address the limitations in taking various features into account, features are defined from results in candidate generation step. Although it adapts a few features, it shows that the proposed model could find the answer without any external resources.

The rest of this paper is organized as follows. Section 2 introduces the approaches refer to the units of to be transliterated and reviews the related works. Section 3 explains the proposed machine transliteration model. In section 4, the experiment and the results are shown and finally Section 5 draws conclusions.

II. Related works

Transliteration is a process that takes a character string in source languages and generates a character string in the target language. It can be seen as two levels: segmentation of the source string into transliteration unit; and transliterate the source language transliteration unit to the target language transliteration unit [10]. Before the review associated to the proposed model, it needs to explain the approach relative to the units to be transliterated. Machine transliteration can

H-J. Song and S-B. Park are with the Department of Computer Science and Engineering, Kyungpook National University, 702-701 Daegu, South Korea. Email: {hjsong, sbpark}@sejong.knu.ac.kr

[1]From now, transliteration is referred to as forward transliteration

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

be classified into phoneme-based, grapheme-based and their hybrid approach in terms of the units to be transliterated.

A phoneme-based approach is to use phonetic information for machine transliteration. It first converts a name in source language into phonemes and then the phonemes with source language graphemes are converted into a target language name. This approach was used in the early stage of the transliteration since it was in accordance with the definition of transliteration [1], [5]. However, the phoneme-based approach usually produces the conversion errors, which propagate to the next step. It makes difficult to transliterate. A grapheme-based approach tries to directly map the source language graphemes to the target languages graphemes without the phonetic information [6], [7]. Compare to the phoneme-based approach, it achieves good performance since it excludes the conversion errors and can be easily performed. A hybrid approach uses a combination of a grapheme-based approach and a phoneme-based approach [4]. This approach is introduced because transliteration is a complex process that does not only rely on the phoneme or grapheme.

Machine transliteration based on the statistical model recently adapt the statistical machine translation technique. Many studies [8], [9] especially used phrase-based statistical machine translation system on transliterating proper names. There showed that a machine transliteration system could be built from an statistical machine translation system whose performance is comparable to state-of-the-art systems designed to transliterate.

Each approach has its own advantages and disadvantages. However, most of the approaches suppose that the object of transliteration, name, comes from only the source language. It is difficult to generate candidates when names are derived from non-source languages.

The web search machine transliteration studies concentrate on the answer search step, not the candidate generation step. Zhou [13] mined the frequency of web pages for ranking candidates. Hong [16] measured the proximity between a source name and candidates and selected the answer. They showed that the web search method were useful to find the appropriate candidate.

In answer search, most machine transliteration systems were based on ranking to find the correct transliteration. They especially used machine learning techniques like a support vector machines or maximum entropy model for ranking candidates. In order to use the machine learning, it needs to define features to represent the relevance between source languages names and target languages candidates. Oh [12] used a lot of features, which unrelate to the relevance. In addition, it needs the external resources to implement features, i.e, pronunciation dictionaries.

This paper propose a hybrid method which combines statistical model on grapheme-based and web search method. By searching web documents, it handles proper names come from various languages. With a few features resulted from the generation step, it could ranks candidates without external resources.

III. Transliteration Model

Fig. 1 shows the overall structure of the proposed model. The proposed model consists of three phrases. In the first phrase, the statistical model is trained with training data which contain entries mapping English names to their respective Korean transliteration. The next phrase, given the test data, candidates are generated based on the statistical model. After generating candidates, words which are likely to be an answer are added from web documents using web search engine. Finally, in a third phrase, a search model ranks the candidates using a ranking function and selects the most appropriate answer. The processes related to candidate generation are called 'Candidate generation' and the remainder are named 'Answer search'.

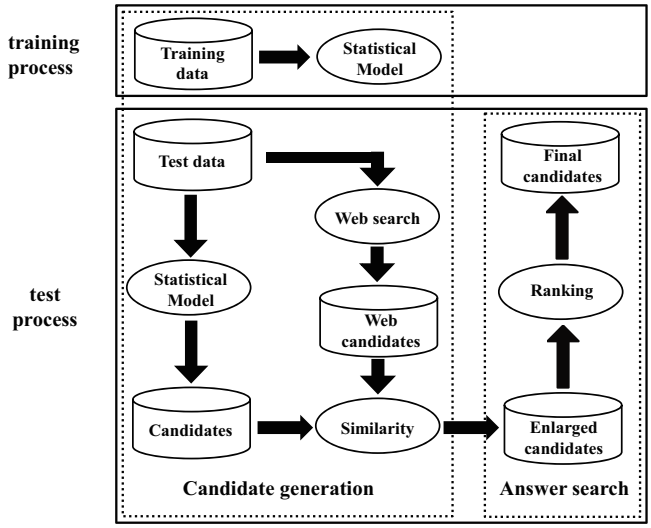

Fig. 1. The proposed system structure

A. Candidate generation

1) Statistical model: Machine transliteration can be regarded as a noisy channel problem. For a given an English name E as the observed channel output, one finds the most likely Korean name K that maximizes $P(K|E)$. Using Bayes' rule, we can formulate the process as Equation 1. That is, the most appropriate Korean name is obtained by

$$K^* = \arg\max_K P(K|E) = \arg\max_K \frac{P(E|K)\,P(K)}{P(K)} \quad (1)$$

Since $P(E)$ is constant for the given K, it can be rewritten as Equation 2:

$$K^* = \arg\max_K P(E|K)P(K) \quad (2)$$

Here, $P(E|K)$ is translation model and $P(K)$ is the language model.

In order to segment the source string into transliteration unit, this paper takes a grapheme-based approach. This helps to minimize errors from the phoneme conversion procedure. As a result, names are easily decomposed into characters(e_i) and Korean graphemes(k_n) respectively without any resources.

$$E = e_1 e_2, ..., e_i$$

$$K = k_1 k_2, ..., k_n$$

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

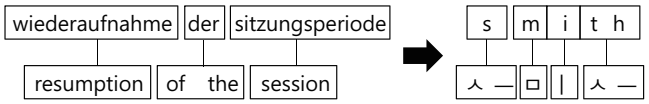

Fig. 2. The relationship between translation and transliteration

With decomposed characters and graphemes, Equation 2 is rewritten as

$$K^* = \underset{k_1 k_2 ... k_n}{\arg\max} P\left(e_1 e_2 ... e_i | k_1 k_2 ... k_n\right) P\left(k_1 k_2 ... k_n\right) \quad (3)$$

To calculate Equation 3, this paper uses Moses [11], a well-known phrase-based statistical machine translation tool. Moses automatically trains translation models for any language pairs with only a collection of parallel corpora. It consists of well-known natural language processing tools and showed the state-of-the-art performance.

Moses is originally designed to deal with the machine translation. In order to use Moses for machine transliteration, the conversion that translation to transliteration is needed. First, the unit of translation is changed from words to characters. Second, the alignment between words should be converted into characters. Fig. 2 shows the example in terms of conversion processes and the relationship between translation and transliteration.

2) Web search: The focus of the candidate generate model is to obtain translation probabilities from a bilingual training corpus. It is regarded as the process that extracts phonetic phenomena from the training pairs and transliterates as the name with the universal phonetic phenomenon. Based on the statistical model, it could generates candidates not same but similar to an answer.

However, it is deficient to generate candidates only with the statistical model. Normally, a language uses proper nouns from various languages. There are a number of names and terms come from not only the source language but also non-source languages. Let us consider the name 'Warsaw' which is the largest city in Poland. Although it is written in English, its origin comes from the Polish. Based on the statistical model, it transliterates as the name relevant to the English not the name corresponding to Polish because the training corpus are composed of the majority of English pairs. That is, the probability that the name 'Warsaw' transliterates as '바르샤바' corresponding to Polish are very low. It means that the statistical model does not generate candidates related to an answer or needs to produce many candidates.

In different way to generate candidates, considers the rule-based method. If an English name is given, the rule-based model detects the origin of word and transliterates with conversion rules corresponding to the origin. However, it is difficult to detect the origin of name, even if the context information is given e.g., the name 'Henry'. A rule-based method does not apply to generate candidates simply.

As a solution of this problem, this paper incorporates a web search method into the candidate generation model, which extracts words from web documents and then adds them to existing candidates. The assumption of candidate generation with web is that relevant transliterations will more frequently appear in WWW documents. With searching web documents, it is possible to generate words which are related to what users commonly used and are similar to an answer.

The web search method is executed as follows. First, it searches documents from the web using the source language name only. At this time, it restricts that documents are written by target language. Second, titles and snippets are extracted from the retrieved documents and then nouns are selected from them with a morphological analyzer since the unit of transliteration, name, is related to noun. Finally, it chooses just few candidates by calculating the similarity with candidates which are generated in statistical model, since it prevent the web search method from generating a lot of different candidates. The similarity is defined as

$$similarity\left(x, y\right) = \alpha \frac{F\left(y\right)}{\arg\max_y F\left(y\right)} + \left(1 - \alpha\right)\left(1 - \frac{ED\left(x, y\right)}{ML\left(x, y\right)}\right) \quad (4)$$

where x is a candidate generated by statistical model and y is a noun extracted from retrieved documents. $F\left(x\right)$ is the frequency of x in retrieved documents, $ED\left(x, y\right)$ is the edit distance between the word x and y. $ML\left(x, y\right)$ returns the longest length between the word x and y for normalizing the value. α is the ratio between the statistical model and the web search. Due to the characteristics of Korean, the unit of functions are set to a character except $F\left(x\right)$.

B. Answer search

The focus of the answer search phrase is to find the correct transliteration from candidates. In this paper, it is regarded as the ranking that candidates related to an answer have high value. On the contrary, candidates unrelated with a correct one have low value.

In order to ranking the candidates, it needs to define an ranking function which determines the plausibility of the candidates with calculating their possibility to be answer. Let H be a set of generated candidates and h_i be the ith candidate of source word and h correct be the answer. X is the feature vectors and x_i is a feature vector of h_i. A ranking function is defined as Equation 5 by ranking $h_{correct}$ higher and the others lower [12].

$$rank\left(x_i\right) : X \rightarrow \{r : r \text{ is ordering of } h_i \in H\} \quad (5)$$

For ranking candidates, this paper uses SVMs, a machine learning based ranking function. For given training data $D = \{(x_1, y_1), ..., (x_n, y_n)\}$, where $x_{correct}$ is a positive sample $(y_{correct} = positive)$ and $x_{i \neq correct}$ is a negative sample $(y_{i \neq correct} = positive)$, the SVMs assign a value to each candidate (h_i) using

$$SVM\left(\mathbf{x_i}\right) = \mathbf{w} \cdot \mathbf{x_i} + b$$

where w denotes a weight vector. It ranks with the value of $SVM\left(\mathbf{x_i}\right)$ since it determines the relative ordering in H.

In order to ranking, it needs to design features to measure the relevance between a source languages word and a candidate. The scores and rank result is defined from candidate generation as features. Table I shows features.

WR indicates the frequency of word on web documents. It represents how many candidates have used universally.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
FEATURES FOR RANKING CANDIDATES

Feature	Explanation
WF	Frequency of word
SR	Rank of statistical model
SP	Value of statistical model

SR and SP are the rank and the value of statistical model respectively. They refer to the suitability and the relative distance.

While the proposed features looks like simple, they have little semantic relation on each others. In addition, these can be extracted from candidate generation step so that it is easily implemented.

IV. EXPERIMENTS

A. Experimental setup

For English-to-Korean transliteration, we used the English-Korean bilingual data which is taken from the National Institute of the Korean Language. The data originally contain 10,373 person names and 12,583 place names including non-ASCII characters. Among them, 18,186 distinct pairs between English and Korean are used in experiment because of non-ASCII characters. The performances of the method are computed using five-fold cross-validation.

In this paper, the machine transliteration performance is evaluated for each step. To evaluate the candidate generation step, coverage are used which measures whether the proposed method generates candidates which contain the answer or not. In search answer, accuracy was used. The coverage and accuracy are calculated as follows.

$$Coverage = \frac{1}{N}\sum_{i=1}^{N} I\left(X_i, Y_i\right)$$

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} 1, & \text{if } \exists X_{i,j} : X_{i,j} = Y_i \\ 0, & \text{otherwise} \end{cases}$$

Note that N is equal to the number of test instances, X_i and Y_i are candidates which are generated from proposed method and answer respectively. $I\left(X_i, Y_i\right)$ is a indicate function that if X_i contains Y_i, returns 1, otherwise returns 0.

To investigate the effect of the proposed method in candidate generation step, statistical model and web search are implemented as baselines. In answer search step, the proposed method is compared with two baselines that one is the random selection and the other is the web frequency selection.

B. Experimental results

Before comparing the proposed method with baseline methods, this paper attempts to evaluate the propriety of the data set. It assesses how many words which comes from non-English languages exist in data set. It is evaluated with a rule-based method based on Romanization notation[2], the primary principle of English-to-Korean transliteration. Table II shows the result.

[2]See http://korean.go.kr/09_new/dic/rule/rule_roman_0101.jsp for more information.

TABLE II
THE PERFORMANCE OF RULE-BASED METHOD

Candidate size	Coverage
136.681 $\pm$ 58	0.2459 $\pm$ 0.002

TABLE III
PARAMETERS AND VALUES IN MOSES

Parameter	value
Language Model Smoothing	Kneser-Ney
Language Model N-Gram Order	3
Maximum Phrase Length Phrase	3
Alignment Heuristic	grow-diag-final
Reordering	msd-bidirectional-fe

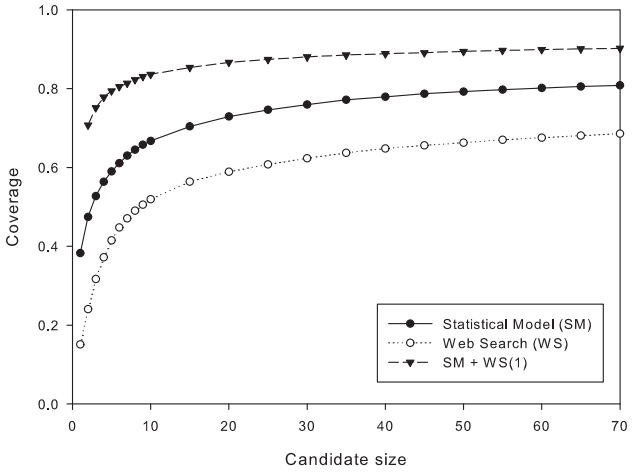

Fig. 3. The coverage according to the candidate size

As shown in Table II, although the rule-based method generated about 130 candidates on averages toward one English name, its coverage is only 24%. It implies that the data set contains many words which comes from non-English language. It is also shown that only rule-based approach could not generate proper candidates.

Next experiment evaluates the performance of candidate generation step. It performs with changing the size of candidates from 1 to 70. In order to implement the statistical model, this paper sets the value of the Moses parameter as table III. For the web search method, Google search engine is used with restricting that the crawled pages are up to 100 pages. Then, nouns are extracted from crawled documents with Hangul Analysis Module (HAM) [15]. The parameter α in equation 4, the ratio between the statistical model and web search, set to 0.3 since it shows the best performance with this value. Furthermore, only one candidate is enlarged from the web search method. Fig. 3 depicts the coverage change according to the candidate size.

Statistical Model (SM) and Web Search (WS) in this graph are the baseline methods explained in Section 4.1 and the SM + WS is the proposed method. Variances are omitted since they are under 0.001 in all experiments. The performance of baselines are from 0.2 to 0.7 which is higher than the performance of the rule-based method. On the other hand, the performance of SM + WS achieves from 0.7 to 0.9 corresponding to candidate size, which is higher 0.2

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

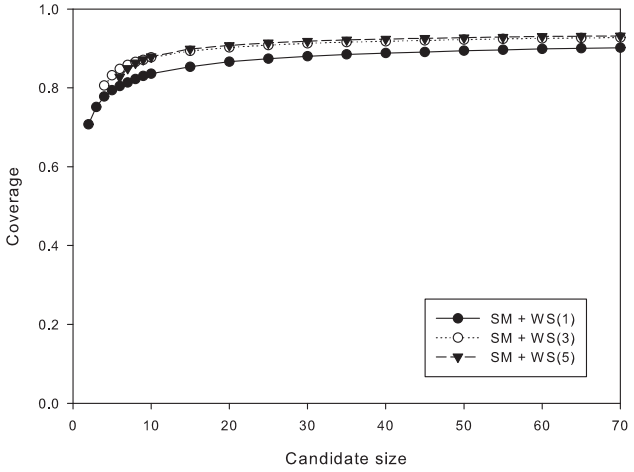

Fig. 4. The coverage according to the candidate size

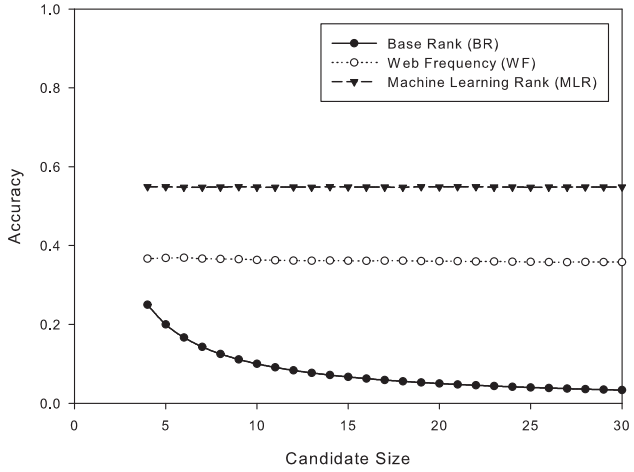

Fig. 5. The top-1 accuracy according to the candidate size

minimum, and 0.6 maximum than baselines. It implies that the proposed method could succeed in generating candidates which could not be created by the statistical model or web search model. It is shown that the coverage of the proposed model monotonically rises up to the candidate size of 20 and then it almost gets flat. The reason why the performance of proposed method is that the similarity between the statistical model and web search are helps to generate few candidates that are alike to the answer. First, by reflecting the frequency, the proposed method eliminates compound words that contain the answer word. Second, it excludes candidates that unrelated to answer using the phonological similarity.

To identify how the number of candidates from web affects the performance, the number of expanded candidates set to from 1 to 5 under the same statistical model condition. Figure 4 shows the experimental results.

The number in parenthesis indicates the number of added candidates from web documents. The result shows that the more the proposed method expanded candidates, the more it covers candidates related to answer. However, there is little difference though the number of candidates is changed. This shows that the similarity between candidates and web nouns is well defined to help generate candidates.

To evaluate the answer search process, SVM^{rank} [14] are used as ranking function and assess the accuracy. The candidates for ranking are set up whose performance is the best at candidate generate step, that is, they are generated by statistical model and add three candidates from the Web with set to the parameter of alpha in similarity as 0.3. Fig. 5 shows the result obtained with the top-1 accuracy with various size of candidates.

The accuracy of the random selection is regarded as the lower bound since it is selected randomly. As the size of candidate increases, the accuracy of the random selection monotonically decreases, on the other hand, the accuracy of web frequency and the proposed method shows the stable situation. This is due to web, candidates are strongly influenced by the result of web in candidate generation step. Above all the accuracy of the proposed method is higher than the web frequency method. It implies that the proposed features are useful for ranking transliteration candidates without other resources. In the future, this study will investigate the use of other features or methods to get higher performance when the size of candidate increases.

V. Conclusion

In this paper, we have proposed a hybrid machine transliteration method, which combines statistical model and web search for transliterating names in various languages. First of all, the proposed method generated candidates based on the phrase-based statistical machine translation model. In order to make up for generating words comes from non-source languages, the web search method used which adds candidates from crawled web documents. As a result, it was able to gain as many candidates as possible, which might contain the correct transliteration. In answer search, this paper formulated searching an answer as ranking. The selection of the most correct transliteration candidate is transformed into choosing the highest value and support vector machine with a few meaningful features is adopted for the ranking. The experimental results showed that the proposed method achieved higher coverage and accuracy than baseline methods. The reason why the proposed method outperforms baseline models is that it considers names originated from various languages with web.

It is generally believed that the phonetic information is useful for transliteration. Thus, our future work will be to extend the model by adding the phonetic information. It is also needed to improve the accuracy up to coverage.

Acknowledgment

This work was supported by the IT R&D program of MKE/IITA. [Development of a Cognitive Planning and Learning Model for Mobile Platforms].

References

[1] Kevin Knight and Jonathan Graehl, “Machine transliteration”, *Computational Linguistics*, vol. 24, no. 4, pp. 128-135, 1997.
[2] Haizhou Li, A kumaran, Vladimir Pervouchine and Min Zhang, “Report of NEWS 2009 Machine Transliteration Shared Task”, In *proceedings of ACL-IJCNLP 2009 Named Entities Workshop*, pp. 1-18, 2009.

[3] Guo-Wei Bian and Hshi-Hsi Chen, "Cross-language information access to multilingual collections on the internet", *Journal of American Society for Information Science*, vol. 51, no. 3, pp. 281-296, 2000.
[4] Jong-Hoon Oh, Key-Sun Choi and Hitoshi Isahara, "A Machine Transliteration Model Based on Correspondence between Graphemes and Phonemes", *ACM Transcations on Asian Language Information Processing*, vol. 5, no. 3, pp. 185-208, 2006.
[5] Bonnie Glover Stalls and Kevin Knight, "Translating names and technical terms in Arabic text", In *proceedings of the Workshop on Computational Approaches to Semitic Languages*, pp. 34-41, 1998.
[6] Byung-Ju Kang and Key-Sun Choi, "Automatic Transliteration and Back-transliteration by Decision Tree Learning", In *proceedings of the 2nd International Conference on Languages Resources and Evaluation*, pp. 1135-1141, 2000.
[7] In-Ho Kang and GilChang Kim, "English-to-Korean Transliteration using Multiple Unbounded Overlapping Phoneme Chunks", In *proceedings of the 18th International Conference on Computational Linguistics*, pp. 418-424, 2000.
[8] David Matthews, "Machine Transliteration of Proper Names", Masters Thesis, School of Informatics, University of Edinburgh, 2007.
[9] Andrew Finch and Eiichiro Sumita, "Phrase-based Machine Transliteration", In *Proceedings of International Joint Conference on Natural Language Processing*, pp. 13-18, 2008.
[10] Haizhou Li, Zhang Min and Su Jian, "A Joint Source-Channel Model for Machine Transliteration", In *Proceeding of the 42nd Annual Meeting on Association for Computational Linguistics*, pp.159-166, 2004.
[11] Phillpp Koehn, Hiue Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin and Evan Herbst, "Moses: Open Source Toolkit for Statistical Machine Translation", In *Proceedings of 45th Annual Meeting of the Computational Linguistics, Demo and poster Sessions*, pp. 177-180, 2007.
[12] Jong-Hoon Oh and Hitoshi Isahara, "Machine Transliteration Using Multiple Transliteration Engines and Hypothesis Re-Ranking", In *Proceedings of the 11th Machine Translation Summit*, pp. 353-360, 2007.
[13] Yilu Zhou, Feng Huang and HsinChun Chen, "Combining probability models and web mining models: a framework for proper name transliteration", *Information Technology and Management*, vol. 9, no. 2, pp. 91-103, 2007.
[14] Thorsten Joachims, "Training Linear SVMs in Linear Time", In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*, 2006.
[15] Seung-Shik Kang, "Korean Morphological Analysis and Information Retrieval", *Hong-Reung publisher*, 2002.
[16] Gumwon Hong, Min-Jeong Kim, Do-Gill Lee and Hae-Chang Rim, "A Hybird Approach to English-Korean Name Transliteration", In *Proceedings of the 2009 Named Entities Workshop*, pp. 108-111, 2009.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)