J. Software Engineering & Applications, **2009, 2: 122-126**
doi:10.4236/jsea.2009.22018 Published Online July 2009 (www.SciRP.org/journal/jsea)

Transliterated Word Identification and Application to Query Translation Mining

Jing Zhang[1], Lei Guo[2], Meiling Zhou[2], Jianmin Yao[2]

[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China; [2]School of Computer Science and Technology, Soochow University, Suzhou, China.
Email: zhjing@scut.edu.cn, jyao@suda.edu.cn

Received February 12th, 2009; revised April 28th, 2009; accepted May 4th, 2009.

ABSTRACT

Query translation mining is a key technique in cross-language information retrieval and machine translation knowledge acquisition. For better performance, the queries are classified into transliterated words and non-transliterated words based on transliterated word identification model, and are further channeled to different mining processes. This paper is a pilot study on query classification for better translation mining performance, which is based on supervised classification and linguistic heuristics. The person name identification gets a precision of over 97%. Transliterated word translation mining shows satisfactory performance.

***Keywords*:** *Transliteration, Query Classification, Supervised Learning, Translation Mining*

1. Introduction

For both cross-language information retrieval and machine translation knowledge acquisition, translation mining for out-of-vocabulary words is an important module which can help translate named-entities, organization and location names, book and movie titles, technical terms, and newly-coined words that are not included in the dictionary.

The web is a rich mineral for translation mining based on co-occurrence statistics. Related researches [1,2,3] use web search engines to get the web snippets for translation and query co-occurrence. Researches [4,5] make study on how to obtain the effective web pages that include both the query and the translation. Besides the co-occurrence statistics, natural language processing techniques such as word alignment is also utilized in recent research work. This paper is a further endeavor to query classification for higher translation mining accuracy.

In past researches, all query terms go through the same process for translation mining, which omits the difference between transliterated words and non-transliterated words. But in fact, general words such as "翻译模型" (translation model) and transliterated words such as "巴拉克·胡赛因·奥巴马" (Barack Hussein Obama) should follow some different mining channels so that the results can achieve higher accuracy. This paper makes an endeavor to an automatic query classification method so that transliterated words can be separated from general query terms.

In present researches on query translation mining, transliterated words are not separated from non-transliterated words. This method leads to a compromised solution in the modeling. [1,6] propose a solution applicable to both transliterated and non-transliterated words, which improved the performance on the whole, but lowered the performance for specialized words.

In transliteration study, [6,7,8] select the most probable translation from series of candidate transliterations. Other transliteration models include [9,10]. Because it's not feasible to identify whether the word is transliterated or not, the work cannot be combined in natural language processing systems up to now.

A method is proposed in this paper to decide whether the query word is a transliterated word or not, which utilizes a unigram-based transliteration statistics plus some heuristic rules. The experiment shows a precision over 97%.

The next section of the paper describes the unigram-based transliteration identification modeling based on a supervised-learning process. The second section describes the experimental results and analysis of the solution. The last section concludes the method and describes future works in the field.

2. A Unigram-Based Transliteration Identification Model

From observation, we can see that the wording of the Chinese transliterated words follows some academic traditions and has good characteristics for statistical study and supervised learning. Two models are proposed from two perspectives and then integrated for better performance on transliterated word identification.

2.1 Transliteration Features of Chinese Characters

Followed are some related concepts for further modeling of the transliteration identification.

Definition 1. Transliteration characters. The Chinese characters that are used in transliterations are called transliteration characters. A national standard for name transliteration exists in China as in [11], which specifies the rules for character selection in name transliteration.

For example, the character "斯" is a transliteration character. While another character, "撕", is not seen in transliterated words, so it's not a transliteration character.

Definition 2. Transliteration probability of a Chinese character. The transliteration probability of a character refers to the probability that the character occurs in transliteration words. The definition is as follows:

$$TP(c) = \frac{the\ number\ of\ \ c\ in\ transliterated\ \ words}{the\ number\ of\ \ all\ the\ transliterated\ character} \quad (1)$$

where, c is a Chinese character. For example, the character "斯" has been seen in transliterated words n_1 times in the running corpus, and the number of transliterated characters is n_2 in the corpus. So $TP(斯)=\frac{n_1}{n_2}$. We use the person name corpus as the training corpus in our experiment.

Based on the above definitions, two models to identify the transliteration words are proposed in the next section.

2.2 Models and the Algorithm for Transliteration Word Identification

Model 1: Counting the number of transliteration characters. Compare the transliterated words like "斯坦福" (Stanford), "克林顿" (Clinton), with non-transliterated words like "星期天" (Sunday), "出版社" (press), we can see that some Chinese characters are frequently seen in transliterated words, e.g. "克" and "斯", which are called transliteration characters. The word "斯坦福" (Stanford) contains 3 transliteration characters, while the word "星期天" (Sunday) contains no transliteration characters. Based on this observation, the first transliteration word identification model is proposed based on the percentage of transliteration characters in the word, as follows:

$$\mathrm{PTC}(w)=\frac{\text{number of transliteration characters in the word } w}{\text{number of characters in the word } w} \quad (2)$$

Based on supervised learning decision, when the PTC (w) is above a threshold, we can decide that the word is a transliterated word. An empirical threshold θ_1 in the following experiment is set at 50.001%.

Model 2: Averaging the transliteration probability of characters. A second model is built based on the average of transliteration probability of all characters in the given Chinese word, defined as follows:

$$A\mathrm{TC}(w)=\frac{\sum_{\text{for all characters } c_i \text{ in } w} TP(c_i)}{number\ of\ characters\ in\ w} \quad (3)$$

The definition of $TP(c_i)$ is given in Equation (1). Similarly, there exists a threshold θ_2 for $ATC(w)$ to decide whether the word is a transliteration or not. The value of the threshold θ_2 is decided by experiments through training, which is $3e^{-5}$ in the following experiment.

Linguistic heuristics: According to observation, some heuristics are applied to enhance the overall performance on basis of combing the two models. The first heuristic is that if the word contains only 1 or 2 characters, Model 1 should be used. For example, the word "卓娅" contains 2 characters, and we should used model.

If the word contains more than 2 characters, use Model 1 first. If Model 1 returns true, then w is a transliterated word. If Model 1 returns false, use Model 2 to make a second identification.

Based on the two models above and the observations, the transliteration word identification process is proposed as follows:

PROCEDURE: Transliteration word identification
BEGIN PROCEDURE
IF the length of the word w is less than 2
THEN use model 1,
IF $PTC(w) > \theta_1$, **RETURN TRUE**
ELSE RETURN FALSE
ELSE IF the length of word w is more than 2
THEN use model 1 first,
IF $PTC(w) > \theta_1$, **RETURN TRUE**
ELSE IF $PTC(w) > \theta_1$,
THEN use model 2 secondly
IF $ATC(w) > \theta_2$, **RETURN TRUE**
ELSE RETURN FALSE.
END PROCEDURE

3. Query Translation Mining from Search Engine Snippets

Taking translation mining system flow is as follows: First, the source Chinese query is sent to a search engine to retrieve Chinese documents. Second, the relevant topic words, which are hint words for the subject or topic of the query, are extracted from the returned snippets. Third, the source query together with the translations of the topic words are sent to search engine again to obtain relevant bilingual web snippets. The next step is extracting valid terms from the returned bilingual snippets and the final step is ranking the candidate terms to get the final translation(s).

Briefly, this system consists of three main parts:

1) Bilingual snippets collection. Retrieve the bilingual snippets that contain the source term in Chinese and translation in English from a search engine and download as the bilingual resource. Effective techniques to obtain higher relevant snippets are the basis of translation extraction.

2) Candidate term extraction. Extract valid lexical units and multi-lexical units from the returned snippet set in Step 1. It is not a straightforward work. Firstly, Chinese texts have no spaces between characters and one snippet with 2 or 3 sentences usually is relatively small compared to authoritative corpora. Secondly, the snippets generally contain OOV terms. Thus term extraction from returned snippets needs specific technical study.

3) Appropriate translation selection. Rank and sort out candidate translations generated in Step 2. The candidate set may be very large, so the most proper translations should be selected from this set.

3.1 Transliteration Model

Proper names, such as person names, place names, etc., compose a large part of OOV terms. Many proper names are translated based on phonetic pronunciations, which we call transliteration. There has been some related work on extracting term translations based on transliteration techniques as in [6,12,13,14,15]. They converted an English name into a phonetic representation, and then transformed the representation sequence into Chinese pin-yin (phonetic sequence) symbols. At last, they translated the pin-yin sequence to a Chinese character sequence. Our transliteration model differs in two aspects. First, the problem is a sort of matching problem. We already have the Chinese candidates and thus do not need to generate the Chinese transliterations. The other difference is, to avoid the double errors from English phonetic representation to pin-yin and from pin-yin to characters, we use a similar idea as in [16,17] to segment an English name into a sequence of syllables, compute the probability between an English syllable and a Chinese character to estimate the possibility. The aim is computing the phonetic similarity for selecting the right translation. First of all, we segment the English term into a sequence of syllables based on heuristic rules and then compute the transliteration cost use the following equation.

$$Trl(s,t) = \frac{P(s,t)}{D(s,t)} \tag{4}$$

where P(s,t) is the co-occurrence probability of s and t which is defined as:

$$P(s,t) \approx \prod_{i=1}^{\min(m,n)} (1-\gamma_1)\, prob(e_i, c_i) \tag{5}$$

where γ_1 is the smoothing weight. $prob(e_i, c_i)$ is the probability between an English syllable e_i and a Chinese character c_i and is computed based dynamic programming from the training corpus contains 37665 proper name pairs. $D(s,t)$ is the number of syllable difference between an English term s and a Chinese candidate t, which is defined as:

$$D(s,t) = \varepsilon + |m-n| \tag{6}$$

Here ε is a decaying parameter, m is the total number of English term syllables and n is the total number of Chinese characters.

In order to improve incorrect transliteration mapping between English syllables and Chinese characters, we combine the forward mapping and backward mapping. The final transliteration cost is defined as:

$$Trl(s,t) = \frac{Trl_F(s,t) + Trl_B(s,t)}{2} \tag{7}$$

where $Trl_F(s,t)$ is the forward transliteration value and $Trl_B(s,t)$ is the backward transliteration value.

4. The Experiment Setup and Result Analysis

Experiments are carried out on transliterated word identification and translation mining. The experiment data and setup is described in this section. And an experiment result analysis is made for further study and its application in query translation mining process.

4.1 Experiment on Transliterated Word Identification

The person name dictionary published by the Xinhua News Agency is used as the training corpus, which contains 37669 person names by transliteration. There are 119329 Chinese characters in the corpus, and 376 different characters.

To test the performance, another different dictionary is used as the test corpus, which contains 106191 transliterated person names.

Because all the words in the dictionaries are person names, some non-transliterated words from a common verbal dictionary are mixed into the test corpus, which contains 12205 items.

The transliteration probability of each character to be part of a transliteration word is first calculated based on the person names from the Xinhua News agency. For each Chinese character t, if it is not in the person name dictionary, the transliteration probability is zero. Or the probability can be calculated based on the following equation,

$$TP(w) = \frac{c(t)}{119329} \tag{8}$$

In which, count(t) is the count of times that the character t occurs in the person name dictionary. 119329 is the total number of Chinese characters in the person names dictionary.

Then, the Model 2 is utilized to calculate the average character probability of each word in the person name dictionary. The minimum value of the probability is set as the threshold value θ_2. Here precision is calculated for evaluation as following.

$$precision = \frac{\#of\ correctly\ classified\ words}{\#of\ words\ for\ classification} \tag{9}$$

At last, the transliteration word identification process is applied based on Model 1, Model 2 and the combination. The performance of the process on the test corpus is shown in the table followed.

4.2 Experiment on Translation Mining from Search Engine Snippets

100 person names are selected randomly from a foreign name dictionary as the test suite. The top N coverage rate, which refers to the ratio of the names whose correct translation is included in the top N mining results, is evaluated. The experiment result is shown in Table 2.

Based on the result above, a comparison of the translation mining is compared with the famous BabelFish translation system. The result is as in Table 3.

False results are underlined in the table. The result shows that our system outperforms BabelFish in transliterated word translation. Result analysis shows that transliteration characters feature is a good feature for transliterated word identification, which can serve as a basis for transliteration word identification. The precision of transliterated word identification becomes low when the precision of non-transliterated word become high. That is because we choose some characters that distinguish between transliterated word and non-transliterated word as transliterated characters. Our approach has failed when we encounter the word such as "巧克力" (chocolate). These words are not transliterated by using standard transliterated characters.

Table 1. Transliterated and non-transliterated word identification performance

Model	classification	Sum	# words classified true	# words classified false	Precision
Model 1	Transliterated word	106191	104197	1994	98.122%
	non-transliterated words	12205	280	11925	97.706%
Model 2	Transliterated word	106191	105939	252	99.763%
	non-transliterated words	12205	1585	10620	87.014%
Model 1& model 2 combined	Transliterated word	106191	104574	1617	98.477%
	non-transliterated words	12205	340	11865	97.212%

Table 2. Top N inclusion rates of the name translation mining from different number of snippets

# returned snippets	Model	Top1	Top3	Top5	Top10
50	Transliteration	52.1%	68.9%	72.5%	78.3%
	+ frequency and distance	58.1%	77.5%	80.6%	89.1%
100	Transliteration	54.3%	71%	76.8%	81%
	+ frequency and distance	73.6%	87.6%	89.1%	94.6%
150	Transliteration	54.4%	70.7%	76.8%	80.8%
	+ frequency and distance	73.6	82.2%	87.6%	91.5%

Table 3. Comparison of translation mining and machine translation of person names

NO.	Name	Mined transliteration	BabelFish machine translation
1	格兰芬多	GRYFFINDOR	the standard sweet orchid smell are many
2	斯莱特林	Slytherin	Slye tring
3	拉文克劳	revenclaw	Lavin crowe
4	赫奇帕奇	HUFFLEPUFF	Hetch Patch
5	韦斯莱	George Weasley	Wei slye
6	格兰杰	granger	Grainger
7	阿不思	albus	Arab league does not think
8	吉德罗	Dumbledore	Lucky Deluo
9	克林顿	Clinton	Clinton
10	布什	Bush	Bush

5. Conclusions

Translation mining is a key process for lexicon acquisition in cross-language information retrieval, machine translation, etc. For better translation mining performance, a supervised transliteration person name identification process is introduced, which helps classify the types of query lexicon. Concepts of transliteration characters and transliteration probability of a character are proposed. Based on the two concepts, two models to identify a transliteration person name are proposed for a supervised classification algorithm. Experiment results show that our method is highly effective.

REFERENCE

[1] F. Huang and Y. Zhang, "Ming key phrase translations from web corpora," Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, pp. 483−490 ACL, 2005.

[2] P. J. Cheng, J. W. Teng, R. C. Chen, J. H. Wang, W. H. Lu, and L. F. Chien, "Translating unknown queries with web corpora for cross-language information retrieval," in the Proceedings of 27th ACM SIGIR, ACM Press, pp. 146−153, 2004.

[3] C. Y. Lu, Y. Xu, and S. Geva, "Web-based query translation for English-Chinese CLIR," Computational Linguistics and Chinese Language Processing, Vol. 13, No. 1, pp. 61−90, 2008.

[4] M. Nagata, T. Saito, and K. Suzuki, "Using the web as a bilingual dictionary," Proceedings of ACL 2001 Workshop Data-Driven Methods in Machine Translation, pp. 95−102. 2001.

[5] W. H. Lu, L. F. Chien, and H. J. Lee, "Translation of web queries using anchor text mining," ACM Transactions on Asian Language Information Processing (TALIP), Vol. 1, No. 2, pp. 159−172, 2002.

[6] S. Li and H. T. Ng, "Mining new word translations from comparable corpora," COLING 2004 ACL, 2004.

[7] M. L. Zhou and J. M. Yao, "Mining named entity transliterations from comparable corpora," Proceedings of 7th International Conference on Chinese Computing, 2007.

[8] J. Li, "Researching and implementing of English-Chinese transliteration method based on text," Master's degree thesis, Harbin Institute of Technology, 2005.

[9] W. Gao, "Phoneme-based statistical transliteration of foreign names for OOV problem [D]," The Chinese University of Hong Kong, 2004.

[10] P. Virga and S. Khudanpur, "Transliteration of proper names in cross-lingual information retrieval[A]," in Proceedings of the ACI Workshop on Multilingual Named Entity Recognition [C], 2003.

[11] Xinhua News Agency, "Translation name office dictionary of world-wide person name translations," China Translation and Publishing Corporation, 1993.

[12] W. H. Lin and H. H. Chen, "Backward machine transliteration by learning phonetic similarity," in Proceedings of CONLL, Taipei, Taiwan, pp. 139−145, 2002.

[13] T. Lin, C. C. Wu, and J. S. Chang, "Word-transliteration alignment," in Proceedings of ROCLING XV, Hsinchu, Taiwan, pp. 1−16, 2003.

[14] W. Gao, K. F. Wong, and W. Lam, "Phoneme-based transliteration of foreign name for OOV problem," in Proceedings of the first International Joint Conference on Natural Language Processing (IJCNLP), Hainan Island, China, pp. 274−381, 2004.

[15] W. Lam, R. Z. Huang, and P. S. Cheung, "Learning phonetic similarity for matching named entity translations and mining new translations," in Proceedings of 27th International ACM SIGIR Conference on Research and Development in Information Retrieval, the University of Sheffield, UK, pp. 281−288, 2004.

[16] S. Wan and C. M. Verspoor, "Automatic English-Chinese name transliteration for development of multilingual resources," in Proceedings of 36th Annual Meeting of the Association for Computational Linguistics, Montreal, Quebec, Canada, pp. 1352−1357, 1998.

[17] W. H. Lu, J. H. Lin, and Y. S. Chang, "Improving translation of queries with infrequent unknown abbreviations and proper names," Computational Linguistics and Chinese Language Processing, Vol. 13, No. 1, pp. 91−120, 2008.

