Using Accessor Variety Features of Source Graphemes in Machine Transliteration of English to Chinese

Mike Tian-Jian Jiang
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan
tmjiang@iis.sinica.edu.tw

Chan-Hung Kuo **Wen-Lian Hsu**
Institute of Information Science
Academia Sinica
Taipei, Taiwan
{laybow, hsu}@iis.sinica.edu.tw

***Abstract*—This work describes a grapheme-based approach of English-to-Chinese (E2C) transliteration, which includes many-to-many alignment models and conditional random fields using accessor variety (AV) as an additional feature based on source graphemes. Experimental results indicate that the AV of a given English segment can generally improve effectiveness of E2C transliteration.**

***Keywords*—named entity; machine transliteration; direct orthographical mapping; expectation maximization; conditional random fields; accessor variety**

I. Introduction

As a subfield of computation linguistics, transliteration refers the phonetic translation of proper nouns and technical terms across languages. Several terms are used interchangeably in the contemporary research literature for the conversion of names between two languages, such as transliteration, transcription, and sometimes Romanization, especially if Latin scripts are used for target strings [1].

This work adopts the same definition of transliteration as during the NEWS 2009 workshop at ACL-IJCNLP 2009 [2] to narrow down “transliteration” to the conversion of a given name in the source language (a text string in the source writing system or orthography) to a name in the target language (another text string in the target writing system or orthography), such that the target language name must fit three specific requirements as follows:

- Phonemically equivalent to the source name;
- Conforms to the phonology of the target language;
- Matches the user intuition of the equivalent of the source language name in the target language, considering the culture and orthographic character usage in the target language.

Among the numerous applications of transliteration of named entities include machine translation, corpus alignment, cross-language information retrieval, information extraction, and automatic lexicon acquisition. Transliteration modeling approaches can be classified as phoneme-based, grapheme-based, and a hybrid of phoneme and grapheme.

Numerous studies have focused on the phoneme-based approach [3-5]. Assume that *E* denotes an English name and *C* represents its Chinese transliteration. The phoneme-based approach converts *E* into an intermediate phonemic representation *p*, and then converts *p* into its Chinese counterpart *C*. Doing so transforms both the source and target names into comparable phonemes, making it relatively easy to determine the phonetic similarity between the two names. The grapheme-based approach, also known as direct orthographical mapping (DOM), has attracted much attention, too [6-13]. This approach treats the transliteration as a statistical machine translation problem under monotonic constraint, and aims to obtain the bilingual orthographical correspondence directly to reduce the possible errors introduced in multiple conversions. The hybrid approach attempts to utilize both phoneme and grapheme information for transliteration. Oh and Choi [14] proposed a strategy to include both phoneme and grapheme features in a single learning process, and subsequently several investigators compared selections between purely phonemic or graphemic features and mixtures of those features [15-22].

This work presents a grapheme-based approach of English-to-Chinese (E2C) transliteration by using many-to-many alignment (M2M-aligner) [15] and conditional random fields (CRF) [23] with an additional feature of accessor variety (AV) [24]. The remainder of this paper is organized as follows. Section 2 briefly introduces related works involving CRF, M2M-aligner, and AV. Section 3 then explains the concept of transliteration using M2M-aligner and CRF. Next, Section 4 summarizes the experimental results, followed by a discussion. Conclusions are finally drawn in Section 5, along with recommendations for future research.

II. Related Works

A. CRF-based Transliteration

Yang *et al.* [10] proposed a two-step CRF model for direct orthographical mapping (DOM) based machine transliteration, in which the first CRF segments a source word into chunks and the second CRF maps the chunks to a word in the target language. Reddy and Waxmonsky [12] presented a phrase-based translation system that characters are grouped into substrings to be mapped atomically into the target language, which showed how substring representation can be incorporated into a CRF model with local context and phonemic information. Shishtla *et al*. (2009) [7] adopted a statistical transliteration technique that consists of alignment model of GIZA++ [25] and CRF model. Aramaki and Abekawwa [9] who focused on fast decoding and easy implementation also used GIZA++ and CRF for sequential labeling of DOM-based transliteration, while Oh *et al*. [19] applied target grapheme and phoneme to CRF, maximum entropy model [26], and margin infused relaxed algorithm [27].

The approach of this work is similar to the techniques of Shishtla *et al*. and Aramaki and Abekawwa, yet this work focuses on the additional AV feature of CRF and uses M2M-aligner, which will be described in the next section, instead of GIZA++.

B. *M2M-aligner*

Jiampojamarn *et al*. argued that previous work has generally assumed one-to-one alignment for simplicity, but letter strings and phoneme strings are not typically in the same length, so null phonemes or null letters must be introduced to make one-to-one-alignments possible [15]. Furthermore, two letters frequently combine to produce a single phoneme (double letters), and a single letter can sometimes produce two phonemes (double phonemes). For example, the English word “ABERT” with its Chinese transliteration “阿贝特”, which Jiampojamarn *et al*. referred as “phonemes”, is aligned as follows.

A　BE　RT
|　|　|
阿　贝　特

The letters “BE” are an example of the double letter problem which mapping to the single phoneme “贝.” These alignments provide more accurate grapheme-to-phoneme relationships for a phoneme prediction model. Hence the M2M-aligner is for alignments between substrings of various lengths and based on the expectation maximization (EM) algorithm. For more details of the algorithm, readers are encouraged to explore previous works [15-17][28].

Despite the ambiguity between Chinese transliteration and phoneme, the above paragraph of the opinion of Jaimpojamarn *et al*. indicates a particular problem of E2C transliteration, that the training data comprised pairs of names written in source and target scripts lacks explicit grapheme-level alignment. This work uses M2M-aligner as an unsupervised method for generating alignments of the training data, which provide hypotheses of DOM without null graphemes.

C. *Accessor Variety*

Feng *et al.* developed an accessor variety (AV) to evaluate the likelihood that a character substring is a Chinese word [24]. Several works have adopted another measurement method called boundary entropy or branching entropy (BE) [29-35]. This determination is closely related to a particular perspective of *n*-gram and information theory of cross entropy or perplexity. According to Zhao and Kit [36], AV and BE both assume that the border of a potential Chinese word is located where the uncertainty of successive characters increases. That work assumed that AV and BE are discrete and continuous versions, respectively, of the fundamental work of Harris [37]; in addition, AV was adopted as an additional feature of CRF-based Chinese Word Segmentation (CWS). The AV of a string *s* is defined as

$$AV(s) = \min\{ L_{av}(s), R_{av}(s)\} \qquad (1)$$

In (3), L_{av}(s) and R_{av}(s) denotes the number of distinct preceding and succeeding characters, except when the adjacent character is absent due to a sentence boundary; the pseudo-character of the beginning or end of a sentence is then accumulated indistinctly. Feng *et al.* also developed more heuristic rules to remove strings that contain known words or adhesive characters [24]. For a strict meaning of unsupervised features and for simplicity, this work does not include those additional rules.

III. Transliteration using EM and CRF

Unlike previous works of CRF-based transliteration involving DOM [7][9][10][12][19] that usually report only one configuration of CRF and assume the initial alignments of name pairs for training have been prepared by GIZA++ (which often needs external data and is relatively complicated to use) or by human annotators, this section provide an easy-to-reproduce automatic procedure using EM-based M2M-aligner and CRF, along with thorough experiments of different feature sets and context depths for CRF configurations.

A. *CRF Aligment Labeling*

The alignment models maximize the likelihood of the observed (source, target) word pairs by using the EM algorithm. However, the initialization often inhibits the performance of the EM algorithm. To obtain better alignment results, this work sets the “maxX” parameter, referring to the maximum size of sub-alignments in the source side is 8, and set the “maxY” parameter, referring to the maximum size of sub-alignments in the target side is 1, since one of the well-known *a* priori of Chinese is that almost all Chinese characters are monosyllabic, which reflects the situation of “double phoneme” mentioned in Section 2.b. Notably, this work follows the definition of grapheme described by Oh and Choi [38] to prevent from confusion of phoneme, grapheme, character, and letter, that graphemes refer to the basic units (or the smallest contrastive units) of written language: for example, English has 26 graphemes or letters or characters, Korean has 24, and German has 30. Table I shows an example of M2M-aligner results. With aligned training data, a transliteration model is then trained by CRF to generate names in the target language

TABLE I. Example of M2M Alignment

<table>
<tr><th>Source</th><th>Target</th><th colspan="2">M2M-Aligner Result</th></tr>
<tr><td>RANARD</td><td>拉纳德</td><td>R:A|N:A:R|D|</td><td>拉|纳|德|</td></tr>
</table>

from names in the source language. Wapiti [39] is used here as the CRF toolkit. Table II presents an example of training data for CRF alignment labeling, where tags B and I indicate whether or not the character is in the starting position of the chunk.

In the proposed labeling scheme, the following configurations are used in E2C transliteration, as shown in Table III. Additionally, C_i refers to the input characters bound individually to the prediction label at its current position *i*. Consider Table II as an example. If the current position is at label "*B* 纳", features generated by C_{-1}, C_0 and C_1 are "A" "N" and "A", respectively. Notably, a prediction label may either consist of a positioning tag and a Chinese character, or simply be the positioning tag. In particular, while the first, the second, the fifth and the sixth "standard runs" (*i.e.* using only the parallel names provided by the corpus for transliteration task to ensure a fair evaluation) appended Chinese characters to all of the positioning tags, the third and fourth standard runs attached Chinese characters to the tag *B* only, but introduce an additional tag *E* representing the ending position of the chunk.

TABLE II. EXAMPLE OF A CRF LABELING FORMAT FOR E2C TRANSLITERATION

Character	Label
R	*B* 拉
A	*I*
N	*B* 纳
A	*I*
R	*I*
D	*B* 德

TABLE III. CONFIGURATIONS IN TRANSLITERATION OF ENGLISH TO CHINESE

ID	Feature Template	AV	Tag	Chinese Char
1	C_0, C_{-1}, C_1 C_{-2}, C_2 C_0C_1 , $C_{-1}C_0$ $C_{-2}C_1$, C_1C_2	No	*B*, *I*	*B* and *I*
2	C_0, C_{-1}, C_1 C_{-2}, C_2 C_0C_1 , $C_{-1}C_0$ $C_{-2}C_1$, C_1C_2	Yes	*B*, *I*	*B* and *I*
3	C_0, C_{-1}, C_1 C_{-2}, C_2 C_0C_1 , $C_{-1}C_0$ $C_{-2}C_1$, C_1C_2	No	*B*, *I*, *E*	*B*
4	C_0, C_{-1}, C_1 C_{-2}, C_2 C_0C_1 , $C_{-1}C_0$ $C_{-2}C_1$, C_1C_2	Yes	*B*, *I*, *E*	*B*
5	C_0, C_{-1}, C_1 C_0C_1 , $C_{-1}C_0$	No	*B*, *I*, *E*	*B*, *I* and *E*
6	C_0, C_{-1}, C_1 C_0C_1 , $C_{-1}C_0$	Yes	*B*, *I*, *E*	*B*, *I* and *E*

B. CRF with AV

The necessity of AV is primarily on the demand for semi-supervised learning. Since AV can be extracted from large corpora without any manual segmentation or annotation, hidden variables underlying frequent surface patterns of languages may be captured via an inexpensive and unsupervised algorithm such as suffix array. Unsupervised feature selection of AV or similar features has generally improved effectiveness of supervised CWS on cross-domain and unlabeled data [40], and this work consequently considers that AV of un-segmented English names from training, development, and test data might help enhancing E2C transliteration.

This work extends the findings of Zhao and Kit [41] to a unified representation of AV features. The representation accommodates both the character position of a string and the string's likelihood ranking by the logarithm. Formally, the ranking function for a string, *s*, with a score, *x*, counted by AV can be expressed as

$$f(s) = r, if\ 2^r \le x < 2^{r+1} \qquad (2)$$

The logarithm ranking mechanism in (4) is inspired by Zipf's law to alleviate the potential data sparseness of infrequent strings. Rank *r* and the corresponding character positions of a string are then concatenated as feature tokens. To clarify the appearance of feature tokens, Table IV presents a sample representation of AV.

For instance, consider strings with two characters, in which one of the strings "RA" is ranked *r* = 5. Therefore, the column of two-character feature tokens has "R" denoted as *5B* and "A" denoted as *5E*. If another two-character string, "AR," competes with "RD" at the position of "R" with a higher rank of *r* = 5, then *5E* is selected to represent the feature of the token at a certain position. Notably, when string "RA" conflicts with string "AN" at position "A" with the same rank of *r* = 5, the corresponding character position with the ranking of the leftmost string, which is *5E* in this case, is applied arbitrarily.

TABLE IV. EXAMPLE OF CRF TRAINING DATA WITH AV FEATURES

Input	AV Feature					Label
	1 char	2 char	3 char	4 char	5 char	
R	*7S*	*5B*	*4B*	*2B*	*0B*	B拉
A	*7S*	*5E*	$4B_1$	$2B_1$	$0B_1$	*I*
N	*6S*	*5E*	*4E*	$2B_2$	$0B_2$	B纳
A	*7S*	*5E*	*3E*	*2E*	*0I*	*I*
R	*7S*	*5E*	*3E*	$2B_2$	*0E*	*I*
D	*7S*	*2E*	*3E*	*2E*	*0E*	B德

IV. Experiments

A. Data Set and Evaluation Metrics

This section summarizes the experimental results. Experiments were conducted on English-Chinese name pairs of shared tasks NEWS-2009 (NEWS09) [2] and NEWS-2010 (NEWS10) [42]. Evaluation metrics and evaluation scripts of NEWS are adopted as well. The following notation is further assumed:

- N: total number of source words in the test set;
- n_i: number of reference transliterations for i-th name in the test set ($n_i \geqq 1$);
- $r_{i,j}$: j-th reference transliteration for i-th name in the test set;
- $c_{i,k}$: k-th candidate transliteration (system output) for i-th name in the test set ($1 \leqq k \leqq 10$);
- K_i: number of candidate transliterations produced by a transliteration system.

Word Accuracy in Top-1 (ACC), also known as Word Error Rate, measures correctness of the first transliteration candidate in the candidate list produced by a transliteration system. ACC = 1 means that all top candidates are correct transliterations *i.e.* they match one of the references, and ACC = 0 means that none of the top candidates are correct.

$$ACC = \frac{1}{N}\sum_{i=1}^{N}\begin{Bmatrix} 1 \text{ if } \exists\, r_{i,j} : r_{i,j} = c_{i,1}; \\ 0 \text{ otherwise} \end{Bmatrix} \tag{5}$$

Fuzziness in Top-1 (Mean F-score) measures how different, on average, the top transliteration candidate is from its closest reference. F-score for each source word is a function of Precision and Recall and equals 1 when the top candidate matches one of the references, and 0 when there are no common characters between the candidate and any of the references. Precision and Recall are calculated based on the length of the Longest Common Subsequence between a candidate and a reference as

$$LCS\,(c,r) = \frac{1}{2}\left(|c| + |r| - ED(c,r)\right) \tag{6}$$

where ED is the edit distance and |x| is the length of x. For example, the longest common subsequence between “abcd” and “afcde” is “acd” and its length is 3. The best matching reference, that is, the reference for which the edit distance has the minimum is taken for calculation. If the best matching reference is given by

$$4 \tag{7}$$

then Recall, Precision and F-score for i-th word are calculated as

$$R_i = \frac{LCS\,(c_{i,1}, r_{i,m})}{|r_{i,m}|} \tag{8}$$

$$P_i = \frac{LCS\,(c_{i,1}, r_{i,m})}{|c_{i,1}|} \tag{9}$$

$$F_i = 2\frac{R_i \times P_i}{R_i + P_i} \tag{10}$$

Mean reciprocal rank (MRR) measures traditional MRR for any right answer produced by the system, from among the candidates. 1/MRR tells approximately the average rank of the correct transliteration. MRR closer to 1 implies that the correct answer is mostly produced close to the top of the n-best lists.

$$RR_i = \begin{Bmatrix} 1 \text{ if } \exists\, r_{i,j}, c_{i,k} : r_{i,j} = c_{i,k}; \\ 0 \text{ otherwise} \end{Bmatrix} \tag{11}$$

$$MRR = \frac{1}{N}\sum_{i=1}^{N} RR_i \tag{12}$$

MAP$_{\mathrm{ref}}$ measures tightly the precision in the n-best candidates for i -th source name, for which reference transliterations are available. If all of the references are produced, then the MAP is 1. Let’s denote the number of correct candidates for the i -th source word in k -best list as $num\,(i,k)$. MAP$_{\mathrm{ref}}$ is then given by

$$MAP_{ref} = \frac{1}{N}\sum_{i}^{N}\frac{1}{n_i}\left(\sum_{k=1}^{n_i} num(i,k)\right) \tag{13}$$

B. Experimental Results and Discussion

The accuracy and F score between development sets (Dev) and test sets (Test) from NEWS10 and NEWS09 were then compared. Table V and VI list the results of the proposed system.

Many pilot tests have been undertaken with both the training set and the development set, followed by evaluation of the development set for optimizing feature combinations and M2M and Wapiti CRF parameters (with Wapiti’s default value of Gaussian prior as 10,000). Sixes configurations of standard runs are selected from them as representative comparisons. The fifth and the sixes standard runs use only unigram and bigram features within the window size of the context in one character. Comparing the fifth standard run with the sixth standard run clearly reveals that AV features significantly improve performances in terms of the accuracy and F-score on the development sets. The third and the fourth standard runs indicate that increasing the window size of the context and decreasing the variety of prediction labels degrades performances slightly. The first and the second standard runs label Chinese character thoroughly with less specific positioning tags intentionally, to facilitate the trade-off between effectiveness and efficiency of CRF training phases. Notably, the second standard runs on test sets are the only places where AV features fail to improve performances of E2C transliteration.

TABLE V. EVALUATION SCORES OF E2C STANDARD RUNS ON NEWS10 CORPUS

Set	ID	ACC	F-Score	MRR	MAP$_{ref}$
Dev	1	85.38	94.71	91.20	85.38
	2	**94.44**	**98.06**	**96.92**	**94.44**
	3	67.04	87.50	76.78	67.04
	4	81.15	93.47	87.89	81.15
	5	78.04	91.81	85.96	78.04
	6	***86.14***	***95.11***	***91.55***	***86.14***
Test	1	***35.47***	**69.79**	**42.52**	***34.04***
	2	**35.63**	***69.76***	***41.93***	**34.14**
	3	30.60	67.33	37.55	29.29
	4	31.67	67.70	38.25	30.24
	5	34.07	69.13	41.62	32.78
	6	34.53	68.98	41.37	33.06

TABLE VI. EVALUATION SCORES OF E2C STANDARD RUNS ON NEWS09 CORPUS

Set	ID	ACC	F-Score	MRR	MAP$_{ref}$
Dev	1	84.94	94.60	90.85	84.94
	2	**94.06**	**98.01**	**96.68**	**94.06**
	3	67.30	87.70	76.99	67.30
	4	81.60	93.46	88.10	81.60
	5	78.66	92.03	86.16	78.66
	6	***86.88***	***95.31***	***91.94***	***86.88***
Test	1	**69.20**	**87.96**	**77.60**	**69.20**
	2	***67.92***	***87.67***	***76.33***	***67.92***
	3	57.87	83.73	68.30	57.87
	4	61.36	85.42	70.96	61.36
	5	66.26	86.96	75.81	66.26
	6	67.30	87.20	75.94	67.30

Since improvements on the test sets are not as good as expected, NEWS data is then carefully investigated. Despite the possibility of over-fitting because of CRF training parameters, one particular phenomenon has been noticed: the development sets contain phrasal named entities that are unseen in the training sets and unused in the test sets. Specifically, E2C word pairs are occasionally impure transliteration and aligned in very different character lengths, such as the name pair of "COMMONWEALTH OF THE BAHAMAS" and "巴哈马 / 联邦." A few E2C name pairs even consist of pure translations, especially for short Chinese names, such as "ARAL SEA" and "咸 / 海." Such names can lead to noisy alignments during the training phases. Li *et al.* also noted that place and company names are sometimes translated in combination of transliteration and meanings, for example, "VICTORIA FALL" becomes "维多利亚 / 瀑布", and then suggested that directed orthographical mapping can be easily extended to handle such name translations [5]. In fact, the M2M parameter "maxX" of this work has been designed for these phrasal structure to be relatively larger and less symmetrical to the parameter "maxY" than those in previous works that normally set both X and Y to 2 as default values. This work analyzes this phenomenon, which is referred to "semi-semantic transliteration" for convenience, by acquiring NEWS Chinese to English (C2E) back-transliteration corpus.

TABLE VII. CONTRIBUTION RATE ACCORDING TO NUMBER OF FEATURES AND LABELS NEWS10 CORPUS

ID	F_{total}	L	C^{Acc}_{Test}	C^{F}_{Test}	C^{MRR}_{Test}	C^{MAP}_{Test}
1	2,501,328	744	**0.0292**	0.0575	**0.0350**	**0.0280**
2	4,882,872	744	***0.0287***	0.0561	***0.0337***	***0.0275***
3	1,125,744	376	0.0273	**0.0601**	0.0335	0.0261
4	2,322,176	376	0.0275	***0.0588***	0.0332	0.0263
5	2,680,512	1,104	0.0272	0.0552	0.0333	0.0262
6	2,975,280	1,104	0.0275	0.0549	0.0329	0.0263

TABLE VIII. CONTRIBUTION RATE ACCORDING TO NUMBER OF FEATURES AND LABELS NEWS09 CORPUS

ID	F_{total}	L	C^{Acc}_{Test}	C^{F}_{Test}	C^{MRR}_{Test}	C^{MAP}_{Test}
1	2,472,300	738	**0.0571**	0.0725	**0.0640**	**0.0571**
2	4,824,306	738	***0.0547***	0.0710	0.0610	***0.0547***
3	1,113,405	373	0.0517	**0.0748**	0.0610	0.0517
4	2,302,156	373	0.0533	***0.0742***	***0.0617***	0.0533
5	2,651,449	1097	0.0530	0.0695	0.0606	0.0530
6	2,946,542	1097	0.0536	0.0695	0.0605	0.0536

The C2E experiments, however, encounter a serious problem of CRF L-BFGS training requirement in terms of space complexity. Therefore the experimental results of C2E transliterations are incomplete and erroneous, since C2E transliteration with the proposed method produces too many labels and features to train a CRF model with the whole training set. In our experience, even a computer with 24GB memory capacity is insufficient for such training. Similar challenges have been noted in the report of NEWS-2010 and reasoned that C2E transliteration is a one-to-many mapping while E2C is a many-to-one mapping [42].

Cohn *et al.* [43] shows that the time complexity of a single iteration, which is required for a typical CRF training using L-BFGS, is $O(L^2NTF)$, where L is the number of labels, N is the number of sequences, T is the average length of the sequences, and F is the average number of activated features of each labeled clique. Meanwhile, it is still an issue to state the precise bound on the number of iterations required for certain tasks. Moreover, efficient CRF implementations usually cache the feature values for every possible clique labeling of the training data, which leads to a memory space requirement of $O(L^2NTF)$, too. According to those analytical results, this work proposes a contribution rate C of the CRF-based transliteration method. For example, the contribution rate of ACC on the test set $C^{ACC}{}_{Test}$ indicates that for each standard run how many ACC point of the test set a unit of computational effort in terms of $log_2(L^2F_{total})$ can provide, where F_{total} is the number of all activated features, which equals to NTF approximately. Judging by the contribution rates of each run, Table VII and Table VIII shows that the first and the second standard runs are usually more efficient than others, except for F-Score.

V. Conclusions and Future Work

This work approximates a phonological context for E2C transliteration with AV as additional graphemic features. As the standard runs are limited by the use of corpus, most systems are implemented under the direct orthographic map. Experimental results indicate that without an explicit phonemic representation of the English source names, using AV features of a given segment can approximate the local phonological context affecting the rendition of a specific segment in Chinese. This work also suggests appropriate parameters of M2M-aligner and optimal configurations of CRF labeling scheme and context depth for E2C transliteration tasks.

To resolve the limitations of this work, we recommend that future research investigate the feasibility of applying different approaches to recognize semi-semantic transliteration with efficient memory usages.

Acknowledgements

This research was supported in part by the National Science Council under grant NSC 100-2631-S-001-001, and the research center for Humanities and Social Sciences under grant IIS-50-23. Ted Knoy is appreciated for his editorial assistance. The authors would like to thank anonymous reviewers for their constructive criticisms.

References

[1] Jack Halpern, “The challenges and pitfalls of Arabic romanization and arabization,” In Proceedings of the Workshop on Comp. Approaches to Arabic Script-based Lang., 2007

[2] Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang, “Report of NEWS 2009 machine transliteration shared task,” In Proceedings of Named Entities Workshop at ACL 2009.

[3] Kevin Knight and Jonathan Graehl, “Machine Transliteration,” Computational Linguistics, vol. 24, no. 4, 1998, pp. 599-612.

[4] Paola Virga and Sanjeev Khudanpur, “Transliteration of Proper Names in Cross-lingual Information Retrieval,” In Proceedings of the ACL Workshop on Multi-lingual Named Entity Recognition, 2003.

[5] Xue Jiang, Le Sun, and Dakun Zhang, “A Syllable-based Name Transliteration System,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 96-99.

[6] Haizhou Li, Min Zhang, and Jian Su, “A joint source-channel model for machine transliteration,” In the Proceedings of the 42nd ACL Annual Meeting, 2004, pp. 159–166.

[7] Praneeth Shishtla, V. Surya Ganesh, Sethuramalingam Subramaniam, and Vasudeva Varma, “A language-independent transliteration schema using character aligned models at NEWS 2009,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 40-43.

[8] Yang Song, Chunyu Kit, and Xiao Chen, “Transliteration of Named Entity via Improved Statistical Translation on Character Sequences,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 57-60.

[9] Eiji Aramaki and Takeshi Abekawwa, “Fast Decoding and Easy Implementation: Tansliteration as Sequential Labeling,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 65-68.

[10] Dong Yang, Paul Dixon, Yi-Cheng Pan, Tasuku Oonishi, Masanobu Nakamura, and Sadaoki Furui, “Combining a two-step conditional random field model and a joint source channel model for machine transliteration,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 72-75.

[11] Yuxiang Jia, Danqing Zhu, and Shiwen Yu, “A Noisy Channel Model for Grapheme-based Machine Transliteration,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 88-91.

[12] Sravana Reddy and Sonjia Waxmonsky, “Substring-based transliteration with conditional random fields,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 92-95.

[13] Oi Yee Kwong, “Graphemic Approximation of Phonological Context for English-Chinese Transliteration,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 186-193.

[14] Jong-Hoon Oh and Key-Sun Choi, “An Ensemble of Transliteration Models for Information Retrieval,” Information Processing and Management, Vol. 42, 2006, pp. 980-1002.

[15] Sittichai Jiampojamarn, Grzegorz Kondrak and Tarek Sherif, “Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion,” In the Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, Rochester, NY, 2007, pp. 372-379.

[16] Sittichai Jiampojamarn, Aditya Bhargava, Qing dou, Kenneth Dwyer, and Grzegorz Kondrak, “DirecTL: a Language-Independent Approach to Transliteration,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 28-31.

[17] Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing dou, Mi-Young Kim, and Grzegorz Kondrak, “Transliteration Generation and Mining with Limited Training Resources,” In the Proceedings of the 2010 Named Entities Workshop, 2010, pp. 39-47.

[18] Martin Jansche and Richard Sproat, “Named Entity Transcription with Pair *n*-Gram Models,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 32-35.

[19] Jong-Hoon Oh, Kiyotaka Uchimoto, and Kentaro Torisawa, “Machine Transliteration using Target-Language Grapheme and Phoneme: Multi-engine Transliteration Approach,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 36-39.

[20] Oi Yee Kwong, “Phonological Context Approximation and Homophone Treatment for NEWS 2009 English-Chinese Transliteration Shared Task,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 76-79.

[21] Dmitry Zelenko, “Combining MDL Transliteration Training with Discriminative Modeling,” In the Proceedings of the 2009 Named Entities Workshop, 2009, pp. 116-119.

[22] Yang Song, Chunyu Kit, and Hai Zhao, “Reranking with Multiple Features for Better Transliteration,” In the Proceedings of the 2010 Named Entities Workshop, 2010, pp. 62-65.

[23] John Lafferty, Andrew McCallum, Fernando Pereira, “Conditional Random Fields Probabilistic Models for Segmenting and Labeling Sequence Data,” In the Proceedings of ICML, 2005, pp. 591-598.

[24] Haodi Feng, Kang Chen, Xiaotie Deng, and Wiemin Zheng, “Accessor Variety Criteria for Chinese Word Extraction,” Computational Linguistics, vol. 30, 2004, pp. 75-93.

[25] Franz Josef Och, Hermann Ney. “A Systematic Comparison of Various Statistical Alignment Models,” Computational Linguistics, vol. 29, no. 1, 2003, pp. 19-51

[26] Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra, “A maximum entropy approach to natural language processing.” Computational Linguistics, vol. 22, no. 1, 1996, pp. 39-71.

[27] Koby Crammer and Yoram Singer, “Ultraconservative Online Algorithms for Multicalss Problems,” Journal of Machine Learning Research, vol. 3, 2003, pp. 951-991

[28] Eric Sven Ristad and Peter N. Yianilos, “Learning String Edit Distance,” IEEE Transactions on Pattern Recognition and Machine Intelligence, Vol. 20, no. 5, 1998, pp.522-532.

[29] Cheng-Huang Tung and His-Jian Lee, “Identification of unknown words from corpus,” Computational Proceedings of Chinese and Oriental Languages, 1994, pp. 131-145.

[30] Jing-Shin Chang and Keh-Yih Su, “An unsupervised iterative method for Chinese new lexicon extraction,” Computation Linguistics and Chinese language Processing, vol. 2, no. 2, 1997, pp. 97-148.

[31] Paul Cohen and Niall Adams, “An Algorithm for Segmenting Categorical Time Series into Meaningful Episodes,” Advances in Intelligent Data Analysis, 2001, pp. 198-207.

[32] Paul Cohen, Niall Adams and Brent Heeringa, “Voting Experts: An Unsupervised Algorithm for Segmenting Sequences,” Intelligent Data Analysis, vol. 11, no. 6, 2007, pp. 607-625.

[33] Paul R Cohen, B Heeringa and Niall M Adams, “An Unsupervised Algorithm for Segmenting Categorical Timeseries into Episodes,” In the Proceedings of the ESF Exploratory Workshop on Pattern Detection and Discovery, 2002, pp. 49-62.

[34] Jin Hu Huang and David Powers, “Chinese Word Segmentation based on contextual entropy,” In the Proceedings of the 17th Asian Pacific Conference on Language, Information and Computation, 2003, pp. 152-158.

[35] Kumiko Tanaka-Ishii, “Entropy as an Indicator of Context Boundaries: An Experiment Using a Web Search Engine” In the Proceedings of International Joint Conference on Natural Language Processing, 2005, pp. 93-105.

[36] Hai Zhao and Chunyu Kit, “Incorporating Global Information into Supervised Learning for Chinese Word Segmentation,” In the Proceedings of the 10th Conference of the Pacific Association for Computation Linguistics, 2007, pp. 66-74.

[37] Zellig Sabbetai Harris, “Morpheme boundaries within words,” In the Papers in Structural and Transformational Linguistics, 1970, pp. 68-77.

[38] J. H. Oh and K. S. Choi, “An Ensemble of Transliteration Models for Information Retrieval,” Information Processing and Management, Vol. 42, 2006, pp. 980-1002.

[39] Thomas Lavergne, Oliver Cappé and François Yvon, “Practical Very Large Scale CRFs,” In the Proceedings the 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 504-513.

[40] Tian-Jian Jiang, Shih-Hung Liu, Cheng-Lung Sung and Wen-Lian Hsu, “Term Contributed Boundary Tagging by Conditional Random Fields for SIGHAN 2010 Chinese Word Segmentation Bakeoff,” Proceeding of the First CIPS-SIGHAN Joint Conference on Chinese Language Pro-cessing, 2010, pp. 266-269.

[41] Hai Zhao and Chunyu Kit, “Unsupervised Segmentation Helps Supervised Learning of Character Tagging for Word Segmentation and Named Entity Recognition,” In the Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing, 2008.

[42] Haizhou Li, A Kumaran, Min Zhang, and Vladimir Pervouchine, “Report of NEWS 2010 transliteration generation shared task,” In Proceedings of the 2010 Named Entities Workshop, 2010, pp. 1-11.

[43] Trevor Cohn, Andrew Smith, and Miles Osborne, “Scaling conditional random fields using error-correcting codes,” In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA, 2005, pp. 10-17.