Multilingual Transliteration Using Feature based Phonetic Method

Su-Youn Yoon, Kyoung-Young Kim and Richard Sproat
University of Illinois at Urbana-Champaign
{syoon9,kkim36,rws}@uiuc.edu

Abstract

In this paper we investigate named entity transliteration based on a phonetic scoring method. The phonetic method is computed using phonetic features and carefully designed pseudo features. The proposed method is tested with four languages – Arabic, Chinese, Hindi and Korean – and one source language – English, using comparable corpora. The proposed method is developed from the phonetic method originally proposed in Tao et al. (2006). In contrast to the phonetic method in Tao et al. (2006) constructed on the basis of pure linguistic knowledge, the method in this study is trained using the Winnow machine learning algorithm. There is salient improvement in Hindi and Arabic compared to the previous study. Moreover, we demonstrate that the method can also achieve comparable results, when it is trained on language data different from the target language. The method can be applied both with minimal data, and without target language data for various languages.

1 Introduction.

In this paper, we develop a multi-lingual transliteration system for named entities. Named entity transliteration is the process of producing, for a name in a source language, a set of one or more transliteration candidates in a target language. The correct transliteration of named entities is crucial, since they are frequent and important key words in information retrieval. In addition, requests in retrieving relevant documents in multiple languages require the development of the multi-lingual system.

The system is constructed using paired comparable texts. The comparable texts are about the same or related topics, but are not, in general, translations of each other. Using this data, the transliteration method aims to find transliteration correspondences in the paired languages. For example, if there were an English and Arabic newspaper on the same day, each of the newspapers would contain articles about the same important international events. From these comparable articles across the paired languages, the same named entities are expected to be found. Thus, from the named entities in an English newspaper, the method would find transliteration correspondences in comparable texts in other languages.

The multi-lingual transliteration system entails solving several problems which are very challenging. First, it should show stable performance for many unrelated languages. The transliteration will be influenced by the difference in the phonological systems of the language pairs, and the process of transliteration differs according to the languages involved. For example, in Arabic texts, short vowels are rarely written while long vowels are written. When transliterating English names, the vowels are disappeared or written as long vowels. For example *London* is transliterated as *lndn* لندن, and both vowels are not represented in the transliteration. However, *Washington* is often transliterated as *wSnjTwn* واشنطون, and the final vowel is realized with long vowel. Transliterations in Chinese are very different from the original English pronunciation due to the

Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics, pages 112–119,
Prague, Czech Republic, June 2007.

limited syllable structure and phoneme inventory of Chinese. For example, Chinese does not allow consonant clusters or coda consonants except [n,ŋ], and this results in deletion, substitution of consonants or insertion of vowels. Thus while a syllable initial /d/ may surface as in *Baghdad* 巴格达 *ba-ge-da*, note that the syllable final /d/ is not represented. Multi-lingual transliteration system should solve these language dependent characteristics.

One of the most important concerns in a multilingual transliteration system is its applicability given a small amount of training data, or even no training data: for arbitrary language pairs, one cannot in general assume resources such as name dictionaries. Indeed, for some rarely spoken languages, it is practically impossible to find enough training data. Therefore, the proposed method aims to obtain comparable performance with little training data.

2 Previous Work

Previous work — e.g. (Knight and Graehl, 1998; Meng et al., 2001; Al-Onaizan and Knight, 2002; Gao et al., 2004) — has mostly assumed that one has a training lexicon of transliteration pairs, from which one can learn a model, often a source-channel or MaxEnt-based model.

Comparable corpora have been studied extensively in the literature, but transliteration in the context of comparable corpora has not been well addressed. In our work, we adopt the method proposed in (Tao et al., 2006) and apply it to the problem of transliteration.

Measuring phonetic similarity between words has been studied for a long time. In many studies, two strings are aligned using a string alignment algorithm, and an edit distance (the sum of the cost for each edit operation), is used as the phonetic distance between them. The resulting distance depends on the costs of the edit operation. There are several approaches that use distinctive features to determine the costs of the edit operation. Gildea and Jurafsky (1996) counted the number of features whose values are different, and used them as a substitution cost. However, this approach has a crucial limitation: the cost does not consider the importance of the features. Nerbonne and Heeringa (1997) assigned a weight for each feature based on entropy and information gain, but the results were even less accurate than the method without weight.

3 Phonetic transliteration method

In this paper, the phonetic transliteration is performed using the following steps:

1) **Generation of the pronunciation for English words and target words**:

a. Pronunciations for English words are obtained using the Festival text-to-speech system (Taylor et al., 1998).

b. Target words are automatically converted into their phonemic level transcriptions by various language-dependent means. In the case of Mandarin Chinese, this is based on the standard Pinyin transliteration system. Arabic words are converted based on orthography, and the resulting transcriptions are reasonably correct except for the fact that short vowels were not represented. Similarly, the pronunciation of Hindi and Korean can be well-approximated based on the standard orthographic representation. All pronunciations are based on the WorldBet transliteration system (Hieronymus, 1995), an ascii-only version of the IPA.

2) **Training a linear classifier using the Winnow algorithm**:

A linear classifier is trained using the training data which is composed of transliteration pairs and non-transliteration pairs. Transliteration pairs are extracted from the transliteration dictionary, while non-transliteration pairs are composed of an English named entity and a random word from the target language newspaper.

a. For all the training data, the pairs of pronunciations are aligned using standard string alignment algorithm based on Kruskal (1999). The substitution/insertion/deletion cost for the string alignment algorithm is based on the baseline cost from (Tao et al, 2006).

b. All phonemes in the pronunciations are decomposed into their features. The features used in this study will be explained in detail in part 3.1.

c. For every phoneme pair (p_1, p_2) in the aligned pronunciations, a feature x_i has a '+1' value or a '−1' value:

$$x_i = \begin{cases} +1 & \text{when } p_1 \text{ and } p_2 \text{ have the same values for feature } x_i \\ -1 & \text{otherwise} \end{cases}$$

d. A linear classifier is trained using the Winnow algorithm from the SNoW toolkit (Carlson et al., 1999).

3) **Scoring English-target word pair:**

a. For a given English word, the score between it and a target word is computed using the linear classifier.

b. The score ranges from 0 to any positive number, and the candidate with the highest score is selected as the transliteration of the given English name.

3.1 Feature set

Halle and Clements (1983)'s distinctive features are used in order to model the substitution/ insertion/deletion costs for the string-alignment algorithm and linear classifier. A distinctive feature is a feature that describes the phonetic characteristics of phonetic segments.

However, distinctive features alone are not enough to model the frequent sound change patterns that occur when words are adapted across languages. For example, stop and fricative consonants such as /p, t, k, b, d, g, s, z/ are frequently deleted when they appear in the coda position. This tendency is extremely salient when the target languages do not allow coda consonants or consonant clusters. For example, since Chinese only allows /n, ŋ/ in coda position, stop consonants in the coda position are frequently lost; *Stanford* is transliterated as *sitanfu*, with the final /d/ lost. Since traditional distinctive features do not consider the position in the syllable, this pattern cannot be captured by distinctive features alone. To capture these sound change patterns, additional features such as "deletion of stop/fricative consonant in the coda position" must be considered.

Based on the pronunciation error data of learners of English as a second language as reported in (Swan and Smith, 2002), we propose the use of what we will term *pseudofeatures*. The pseudo features in this study are same as in Tao et al. (2006). Swan & Smith (2002)'s study covers 25 languages including Asian languages such as Thai, Korean, Chinese and Japanese, European languages such as German, Italian, French and Polish, and Middle East languages such as Arabic and Farsi. The substitution/insertion/deletion errors of phonemes were collected from this data. The following types of errors frequently occur in second language learners' speech production.

(1) **Substitution**: If the learner's first language does not have a particular phoneme found in English, it is substituted by the most similar phoneme in their first language.

(2) **Insertion**: If the learner's first language does not have a particular consonant cluster in English, a vowel is inserted.

(3) **Deletion**: If the learner's first language does not have a particular consonant cluster in English, one consonant in the consonant cluster is deleted.

The same substitution/deletion/insertion patterns in a second language learner's errors also appear in the transliteration of foreign names. The deletion of the stop consonant which appears in English-Chinese transliterations occurs frequently in the English pronunciation spoken by Chinese speakers. Therefore, the error patterns in second language learners' can be used in transliteration.

Based on (1) ~ (3), 21 pseudo features were designed. All features have binary values. Using these 21 pseudo features and 20 distinctive features, a linear classifier is trained. Some examples of pseudo features are presented in Table 1.

Pseudo-Feature	Description	Example
Consonant-coda	Substitution of consonant feature in coda position	
Sonorant-coda	Substitution of sonorant feature in coda position	Substitution between [ŋ] and [g] in coda position in Arabic
Labial-coda	Substitution of labial feature in coda position	Substitution between [m] and [n] in coda position in Chinese
j-exception	Substitution of [j] and [dʒ]	Spanish/Catalan and Festival error
w-exception	Substitution of [v] and [w]	Chinese/Farsi and Festival error

Table 1. Examples of pseudo features

3.2 Scoring the English-target word pair

A linear classifier is trained using the Winnow algorithm from the SNoW toolkit.

The Winnow algorithm is one of the update rules for linear classifier. A linear classifier is an algorithm to find a linear function that best separates the data. For the set of features X and set of weights W, the linear classifier is defined as [1] (Mitchell, T., 1997)

$$X = \{x_1, x_2, \dots x_n\}$$
$$W = \{w_1, w_2, \dots w_n\} \qquad [1]$$
$$f(x) = \begin{cases} 1 & \text{if } w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n > 0 \\ -1 & \text{otherwise} \end{cases}$$

The linear function assigns label +1 when the paired target language word is the transliteration of given English word, while it assigns label –1 when it is not a transliteration of given English word.

The score of an English word and target word pair is computed using equation [2] which is part of the definition of *f(x)* in equation [1].

$$w_0 + \sum_{i=1}^{n} w_i x_i \qquad [2]$$

The output of equation [2] is termed *the target node activation*. If this value is high, class 1 is more activated, and the pair is more likely to be a transliteration pair. To illustrate, let us assume there are two candidates in target language (t_1 and t_2) for an English word e. If the score of (e, t_1) is higher than the score of (e, t_2), the pair (e, t_1) has stronger activation than (e, t_2). It means that t_1 scores higher as the transliteration of e than t_2. Therefore, the candidate with the highest score (in this case t_1) is selected as the transliteration of the given English name.

4 Experiment and Results

The linear function was trained for each language, separately. 500 transliteration pairs were randomly selected from each transliteration dictionary, and used as positive examples in the training procedure. This is quite small compared to previous approaches such as Knight and Graehl (1998) or Gao et al. (2004). In addition, 1500 words were randomly selected from the newspaper in the target languages, and paired with English words in the positive examples. A total of 750,000 pairs (500 English words $\times$ 1500 target words) were generated, and used as negative examples in the training procedure.

Table 2 presents the source of training data for each language.

	Transliteration pair	Target word
Arabic	New Mexico State University	Xinhua Arabic newswire
Chinese	Behavior Design Corporation	Xinhua Chinese newswire
Hindi	Naidunia Hindi newswire	Naidunia Hindi newswire
Korean	the National Institute of the Korean language	Chosun Korean newspaper

Table 2. Sources of the training data

The phonetic transliteration method was evaluated using comparable corpora, consisting of newspaper articles in English and the target languages—Arabic, Chinese, Hindi, and Korean– from the same day, or almost the same day. Using comparable corpora, the named-entities for persons and locations were extracted from the English text; in this paper, the English named-entities were extracted using the named-entity recognizer described in Li et al. (2004), based on the SNoW machine learning toolkit (Carlson et al., 1999).

The transliteration task was performed using the following steps:

1) English text was tagged using the named-entity recognizer. The 200 most frequent named entities were extracted from seven days' worth of the English newswire text. Among pronunciations of words generated by the Festival text-to speech system, 3% contained errors representing monophthongs instead of diphthongs or vice versa. 1.5% of all cases misrepresented single consonant, and 6% showed errors in the vowels. Overall, 10.5% of the tokens contained pronunciation errors which could trigger errors in transliteration.

2) To generate the Arabic and Hindi candidates, all words from the same seven days were extracted. In the case of Korean corpus, the collection of newspapers was from every five days, unlike the other three language corpora which were collected every day; therefore, candidates of Korean were

generated from one month of newspapers, since seven days of newspaper articles did not show a sufficient number of transliteration candidates. This caused the total number of candidates to be much bigger than for the other languages.

The words were stemmed all possible ways using simple hand-developed affix lists: for example, given a Hindi word c1c2c3, if both c3 and c2c3 are in the suffix and ending list, then this single word generated three possible candidates: c1, c1c2, and c1c2c3.

3) Segmenting Chinese sentences requires a dictionary or supervised segmenter. Since the goal is to use minimal knowledge or data from the target language, using supervised methods is inappropriate for our approach. Therefore, Chinese sentences were not segmented. Using the 495 characters that are frequently used for transliterating foreign names (Sproat et al., 1996), a sequence of three of more characters from the list was taken as a possible candidate for Chinese.

4) For the given 200 English named entities and target language candidate lists, all the possible pairings of English and target-language name were considered as possible transliteration pairs.

The number of candidates for each target language is presented in Table 3.

Language	The number of candidates
Arabic	12,466
Chinese	6,291
Hindi	10,169
Korean	42,757

Table 3. Number of candidates for each target language.

5) Node activation scores were calculated for each pair in the test data, and the candidates were ranked by their score. The candidate with the highest node activation score was selected as the transliteration of the given English name.

Some examples of English words and the top three ranking candidates among all of the potential target-language candidates were given in Tables 4, 5. Starred entries are correct.

English Word	Rank	Candidate	
		Script	Romanization
Arafat	*1	阿拉法特	a-la-fa-te
	2	拉法地奥	la-fa-di-ao
	3	拉维奇	la-wei-qi

Table 4. Examples of the top-3 candidates in the transliteration of English – Chinese

English Word	Rank	Candidate	
		Script	Romanization
Vietnam	*1	베트남	be-thu-nam
	2	베트남측	be-thu-nam-chug
	3	표준어와	pyo-jun-e-wa
Australia	*1	오스트레일리아	o-su-thu-ley-il-li-a
	2	웃돌아	us-tol-la
	3	오스트레일리아에서	o-su-thu-ley-il-li-a-ey-se

Table 5. Examples of the top-3 candidates in the transliteration of English-Korean

To evaluate the proposed transliteration methods quantitatively, the Mean Reciprocal Rank (MRR), a measure commonly used in information retrieval when there is precisely one correct answer (Kandor and Vorhees, 2000) was measured, following Tao and Zhai (2005).

Since the evaluation data obtained from the comparable corpus was small, the systems were evaluated using both held-out data from the transliteration dictionary and comparable corpus.

First, the results of the held-out data will be presented. For a given English name and target language candidates, all possible combinations were generated. Table 6 presents the size of held-out data, and Table 7 presents MRR of the held-out data.

	Number of English named entities	Number of Candidates in target language	Number of total pairs used in the evaluation
Arabic	500	1,500	750,000
Chinese	500	1,500	750,000
Hindi	100	1,500	150,000
Korean	100	1,500	150,000

Table 6. Size of the test data

	Baseline	Winnow	
		Total feature	distinctive feature only
Arabic	0.66	0.74	0.70
Chinese	0.74	0.74	0.72
Hindi	0.87	0.91	0.91
Korean	0.82	0.85	0.82

Table 7. MRRs of the phonetic transliteration

The baseline was computed using the phonetic transliteration method proposed in Tao et al. (2006). In contrast to the method in this study, the baseline system is purely based on linguistic knowledge. In the baseline system, the edit distance, which was the result of the string alignment algorithm, was used as the score of an English-target word pair. The performance of the edit distance was dependent on insertion/deletion/ substitution costs. These costs were determined based on the distinctive features and pseudo features, based on the pure linguistic knowledge without training data. As illustrated in Table 7, the phonetic transliteration method using features worked adequately for multilingual data, as phonetic features are universal, unlike the phonemes which are composed of them. Adopting phonetic features as the units for transliteration yielded the baseline performance.

In order to evaluate the effectiveness of pseudo features, the method was trained using two different feature sets: a total feature set and a distinctive feature-only set. For Arabic, Chinese and Korean, the MRR of the total feature set was higher than the MRR of the distinctive feature-only set. The improvement of the total set was 4% for Arabic, 2.6% for Chinese, 2.4% for Korean. There was no improvement of the total set in Hindi. In general, the pseudo features improved the accuracy of the transliteration.

For all languages, the MRR of the Winnow algorithm with the total feature set was higher than the baseline. There was 7% improvement for Arabic, 0.7% improvement for Chinese, 4% improvement for Hindi and 3% improvement for Korean.

We turn now to the results on comparable corpora. We attempted to create a complete set of answers for the 200 English names in our test set, but part of the English names did not seem to have any standard transliteration in the target language according to the native speaker's judgment. Accordingly, we removed these names from the evaluation set. Thus, the resulting list was less than 200 English names, as shown in the second column of Table 8; (Table 8 *All*). Furthermore, some correct transliterations were not found in our candidate list for the target languages, since the answer never occurred in the target news articles; (Table 8 *Missing*). Thus this results in a smaller number of candidates to evaluate. This smaller number is given in the fourth column of Table 8; (Table 8 *Core*).

Language	# All	# Missing	#Core
Arabic	192	121	71
Chinese	186	92	94
Hindi	144	83	61
Korean	195	114	81

Table 8. Number of evaluated English Name

MRRs were computed on the two sets represented by the count in column 2, and the smaller set represented by the count in column 4. We termed the former MRR "AllMRR" and the latter "CoreMRR". In Table 9, "CoreMRR" and "AllMRR" of the method were presented.

	Baseline		Winnow	
	All-MRR	Core MRR	All-MRR	Core MRR
Arabic	0.20	0.53	0.22	0.61
Chinese	0.25	0.49	0.25	0.50
Hindi	0.30	0.69	0.36	0.86
Korean	0.30	0.71	0.29	0.69

Table 9. MRRs of the phonetic transliteration

		test data			
		Arabic	Chinese	Hindi	Korean
training data	Arabic	**0.61**	0.50	0.86	0.63
	Chinese	0.59	**0.50**	0.80	0.66
	Hindi	0.59	0.54	**0.86**	0.67
	Korean	0.56	0.51	0.76	**0.69**

Table 10. MRRs for the phonetic transliteration 2

In both methods, CoreMRRs were higher than 0.49 for all languages. That is, if the answer is in the target language texts, then the method finds the correct answer within the top 2 words.

As with the previously discussed results, there were salient improvements in Arabic and Hindi when using the Winnow algorithm. The MRRs of the Winnow algorithm except Korean were higher than the baseline. There was 7% improvement for Arabic and 17% improvement for Hindi in CoreMRR. In contrast to the 3% improvement in held-out data, there was a 2% decrease in Korean: the MRRs of Korean from the Winnow algorithm were lower than baseline, possibly because of the limited size of the evaluation data. Similar to the results of held-out data, the improvement in Chinese was small (1%).

The MRRs of Hindi and the MRRs of Korean were higher than the MRRs of Arabic and Chinese. The lower MRRs of Arabic and Chinese may result from the phonological structures of the languages. In general, transliteration of English word into Arabic and Chinese is much more irregular than the transliteration into Hindi and Korean in terms of phonetics.

To test the applicability to languages for which training data is not available, we also investigated the use of models trained on language pairs *different from* the target language pair. Thus, for each test language pair, we evaluated the performance of models trained on each of the other language pairs. For example, three models were trained using Chinese, Hindi, and Korean, and they were tested with Arabic data. The CoreMRRs of this experiment were presented in Table 10. Note that the diagonal in this Table represents the within-language-pair training and testing scenario that we reported on above.

For Arabic, Hindi, and Korean, MRRs were indeed the highest when the methods were trained using data from the same language, as indicated by the boldface MRR scores on the diagonal. In general, however, the MRRs were not saliently lower across the board when using different language data than using same-language data in training and testing. For all languages, MRRs for the cross-language case were best when the methods were trained using Hindi. The differences between MRRs of the method trained from Hindi and MRRs of the method by homogeneous language data were 2% for Arabic and Korean. In the case of Chinese, MRRs of the method trained by Hindi was actually *better* than MRRs obtained by Chinese training data. Hindi has a large phoneme inventory compared to Korean, Arabic, and Chinese, so the relationship between English phonemes and Hindi phonemes is relatively regular, and only small number of language specific transliteration rules exist. That is, the language specific influences from Hindi are smaller than those from other languages. This characteristic of Hindi may result in the high MRRs for other languages. What these results imply is that named entity transliteration could be performed without training data for the target language with phonetic feature as a unit. This approach is especially valuable for languages for which training data is minimal or lacking.

5 Conclusion

In this paper, a phonetic method for multilingual transliteration was proposed. The method was based on string alignment, and linear classifiers trained using the Winnow algorithm. In order to learn both language-universal and language-specific transliteration characteristics, distinctive features and pseudo features were used in training. The method can be trained using a small amount of training data, and the performance decreases only by a small degree when it is trained with a language different from the test data. Therefore, this method is extremely useful for underrepresented languages for which training data is difficult to find.

Acknowledgments

This work was funded the National Security Agency contract NBCHC040176 (REFLEX) and a Google Research grant.

References

Y. Al-Onaizan and K. Knight. 2002. Machine transliteration of names in Arabic text. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, Philadelphia, PA.

Andrew J. Carlson, Chad M. Cumby, Jeff L. Rosen, and Dan Roth. 1999. The SNoW learning architecture. *Technical Report UIUCDCS-R-99-2101*, UIUC CS Dept.

Wei Gao, Kam-Fai Wong, and Wai Lam. 2004. Phoneme based transliteration of foreign names for OOV problem. *Proceeding of IJCNLP*, 374–381.

Daniel Gildea and Daniel Jurafsky. 1996. Learning Bias and Phonological-Rule Induction. *Computational Linguistics* 22(4):497–530.

Morris Halle and G.N. Clements. 1983. *Problem book in phonology*. MIT press, Cambridge.

James Hieronymus. 1995. *Ascii phonetic symbols for the world's languages: Worldbet.* http://www.ling.ohio-tate.edu/ edwards/worldbet.pdf.

Paul B. Kantor and Ellen B. Voorhees. 2000. The TREC-5 confusion track: Comparing retrieval methods for scanned text. *Information Retrieval*, 2: 165–176.

Kevin Knight and Jonathan Graehl. 1998. Machine transliteration. *Computational Linguistics*, 24(4).

Joseph B. Kruskal. 1999. An overview of sequence comparison. *Time Warps, String Edits, and Macromolecules*, CSLI, 2nd edition, 1–44.

Xin Li, Paul Morie, and Dan Roth. 2004. Robust reading: Identification and tracing of ambiguous names. *Proceeding of NAACL-2004*.

H.M. Meng, W.K Lo, B. Chen, and K. Tang. 2001. Generating phonetic cognates to handle named entities in English-Chinese cross-language spoken document retrieval. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop*.

Tom M. Mitchell. 1997. *Machine Learnin*g, McCraw-Hill, Boston.

John Nerbonne and Wilbert Heeringa. 1997. Measuring Dialect Distance Phonetically. *Proceedings of the 3rd Meeting of the ACL Special Interest Group in Computational Phonology*.

Richard Sproat, Chilin. Shih, William A. Gale, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3).

Michael Swan and Bernard Smith. 2002. *Learner English*, Cambridge University Press, Cambridge .

Tao Tao and ChengXiang Zhai. 2005. Mining comparable bilingual text corpora for cross-language information integration. *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 691–696.

Tao Tao, Su-Youn Yoon, Andrew Fister, Richard Sproat and ChengXiang Zhai. "Unsupervised Named Entity Transliteration Using Temporal and Phonetic Correlation." EMNLP, July 22-23, 2006, Sydney, Australia.

Paul A. Taylor, Alan Black, and Richard Caley. 1998. The architecture of the Festival speech synthesis system. *Proceedings of the Third ESCAWorkshop on SpeechSynthesis*, 147–151.