Improving Transliteration Mining by Integrating Expert Knowledge with Statistical Approaches

Ohnmar Htun
Dept. of Management and Information System Science
Nagaoka University of Technology
Japan

Andrew Finch
Multilingual Translation Laboratory
Universal Communication Research Institute
National Institute of Information and Communication Technology (NICT)
Japan

Eiichiro Sumita
Multilingual Translation Laboratory
Universal Communication Research Institute
National Institute of Information and Communication Technology (NICT)
Japan

Yoshiki Mikami
Dept. of Management and Information System Science
Nagaoka University of Technology
Japan

ABSTRACT

This paper contributes a study of methods for integrating human expert knowledge with machine learning approaches for determining phonetic similarity of word pairs. A method is proposed which allows a human to provide a structure for the edit costs that are based around a phonetically-motivated model of phoneme sound groups, and the machine to determine precise values for these costs within two different frameworks based on stochastic edit distance: a method based on one-to-one expectation maximization (EM) alignment and a Bayesian many-to-many alignment approach. A preliminary study is within the context of cross-language word similarity in transliteration mining. The experiments were performed on a Myanmar-English mining task; the principle approach is expected to be most useful for low-resource language pairs, where human expert knowledge can compensate for a lack of data resources. The results show that the approach outperforms baseline systems based on only human knowledge and only on machine learning. This approach showed the choice of edit cost is a strong factor in determining the performance of the edit-distance-based techniques used in these experiments. The learned edit costs consistently outperformed a simple set of plausible costs selected by a human expert. Furthermore, providing a structure to the weights for the machine learning process reduced the number of parameters to be learned simplifying and speeding up the learning task. This method is expected to mitigate issues with data sparseness when learning models for low-resource languages. The reduction in the number of model parameters led to improvements in recall in these experiments, even though the model was considerably smaller, validating the choice of structure.

General Terms

Machine Translation, Transliteration Mining, Information Retrieval, Phonetic Similarity

Keywords

Machine Learning, Transliteration Mining, Cross Language Information Retrieval, Phonetic Similarity, Statistical Approaches, Stochastic Edit Distance

1. INTRODUCTION

Transliteration is the process of converting characters from the scripts of a source language to another by replacing them with approximate phonetic equivalents in the target language script [1]. Thus transliteration preserves the sounds of the syllables in words as far as possible subject to the constraints imposed by the phonetics of the languages involved. A process of transliteration is often used in machine translation to translate proper nouns and technical terms, which are not listed in the bilingual dictionaries and corpora. It has been shown experimentally through a human evaluation by [2] that transliterating out of vocabulary words can be a more effective strategy than simply deleting them from the output.

Transliterated words are also known as loan words or borrowed words, and these words have no genetic relationship between them and the words from which they are derived. Japanese has borrowed many words from Chinese and European languages. For example, "Olympic" and "Platinum" in English are transliterated as "オリンピック" (O-RI-N-PI-KU) and "プラチナ" (PU-RA-CHI-NA) in Japanese and similarly "津波" (tsunami) and "柔道" (judo) in Japanese are loan words in English.

Transliteration mining is the task of extracting transliterated word pairs from parallel or comparable corpora automatically. It is applicable in many applications such as building training data for transliteration generation systems, enhancing lexical coverage for machine translation, and expanding translation resources for cross-language information retrieval (CLIR) systems (in this application transliteration mining is used to handle out-of-vocabulary query words between queries and retrieving documents).

2. MOTIVATION

Although much research and development has already been done in machine translation, there are many remaining challenges in low-resource languages like Myanmar. These challenges often arise because of a lack of bilingual data resources in particular and/or sometimes a lack of any data in general. When data are plentiful, one might expect a statistical approach to transliteration mining to be more effective than other approaches; if parallel corpora are available, good results can be achieved in transliteration of word pairs, but such corpora are still rare for many languages. The research in this paper is motivated by the need to overcome two major problems that are encountered while attempting to mine Myanmar/English transliteration data:

i. Currently, there is only limited bilingual training data available for Myanmar. This data may not be sufficient to obtain accurate estimates of the probabilities used in a statistical alignment process.

ii. The Myanmar writing system is syllable-based where each syllable consists of vowels or consonants and vowels together. A word can be a combination of one or more syllables. There are no delineating features, such as capitalization and spaces to break the words. This leads a number of issues that need to be resolved:

    1. If used directly, the large syllable set size in Myanmar can lead to data sparseness problems when using statistical approaches.

    2. A simple edit-distance based strategy cannot be used to meaningfully compare Myanmar syllables to English characters, since the mapping is not one-to-one.

    3. There is no standard romanization system for Myanmar; if the syllables are to be decomposed or Romanized, the question of how this might be done must be addressed.

This paper focuses on addressing these problems to improve the overall performance of transliteration mining. The proposed method allows a human to provide a structure for the edit costs that is based around a phonetically-motivated model of similar phoneme groups, and a machine to determine precise values for these costs within two different frameworks based on stochastic edit distance: a method based on one-to-one EM alignment [3] and a non-parametric Bayesian many-to-many alignment approach [4].

The remainder of this paper is organized as follows. Section 2 describes the motivation of research; Section 3 surveys prior work on transliteration mining tasks; Section 4 defines the phonetic-based edit grouping system; Section 5 presents the mining methodology; Section 6 gives the experimental data, results, and discussion; Section 7 concludes the paper.

3. PREVIOUS WORK

Traditionally transliteration mining systems have been applied to reasonably large-scale data resources in various language scripts, which have been studied in several prior works. In order to improve the performance of cross-language information retrieval (CLIR), in [5] the effect of integrating transliteration mining and transliteration generation techniques into CLIR was studied. They found that transliteration mining techniques were able to give better results than applying transliteration generation techniques. An experiment was done in the context of Hindi-English and Tamil-English on the standard FIRE 2010 dataset[①]. The transliteration similarity model was built using a W-HMM word alignment model [6] to determine whether document term was a transliteration of the query term. The expectation maximization (EM) algorithm was used to estimate the model parameters and transliteration similarity score of each source and target pair (w_s,w_t) was defined to be log(w_t|w_s). They combined both techniques, but this approach did not produce significantly better results than using transliteration mining alone.

In [7], a generative transliteration model was trained using limited resources by using two methods: phonetic conflation and iterative training of the transliteration model. Phonetic conflation used a Soundex like conflation scheme for English. The experiment tested transliteration from five source languages (Arabic, Chinese, Hindi, Russian, and Tamil) to the target, English. The transliteration model with phonetic conflation gave much improved recall and F-measure in general. The model without phonetic conflation gave improved recall but often at the expense of precision.

Arabic-English transliteration mining using large training and test datasets was performed by applying a graph reinforcement method in [8, 9]. The baseline transliteration mining was trained by using a Bayesian generative model and the alignment of the character pairs was done by using an HMM based aligner [6]. Each source/target character sequence used in the alignment had a maximum length of 3 characters along with their associated mappings into the target language. Although a large amount of training data yielded more correct initial mappings, it tended to increase the errors. A method of graph reinforcement that led to sizable improvement in precision was introduced.

A classical stochastic model that learned a string edit distance function from a corpus of examples was proposed in [3]. Edit weights were learned for four primitive edit operations: identity, insertion, deletion, and substitution (the edit operators that are used in the standard Levenshtein distance [10]). The generative model learned multiple edit paths by using an expectation-maximization (EM) algorithm in an unsupervised manner. The expectation step accumulated expected counts for each edit operation on the training corpus. The maximization step set the model parameter values using these expectations. The total probability of all edit operations beings normalized in the maximization step. This approach is applicable to various applications involving string similarity and was shown empirically to reduce the error with respect to using untrained Levenshtein distance with unit edit costs.

A model of semi-supervised transliteration mining was proposed in [11] which incorporates an explicit model for the generation of non-transliteration pairs (which will be referred to as the noise model). The model classifies unseen pairs by comparing the probabilities assigned by the transliteration sub-model, and the noise sub-model. In EM training, a parameter λ, the prior probability of generating a non-transliteration pair is learned along with model parameters representing the probabilities of each edit operation. Experiments were conducted on four language pairs: English-Arabic, English-Hindi, English-Tamil and English-Russian. Their results show that semi-supervised mining performed much better than an unsupervised approach. The current system is limited to learning unigram character alignment. In this experimental section, their approach is used as a baseline

[①] Forum for Information Retrieval Evaluation (FIRE) http://www.isical.ac.in/~clia/data.html system that will refer to as EMn (since this model is trained using the EM algorithm, and includes a model for the noise). The proposed method extends their model to allow the integration of human knowledge of phonetic features and describe this in detail in the next section.

As an alternative to the EM alignment approach of [11], a non-parametric Bayesian alignment approach was proposed by the authors in [4, 12]. This Bayesian approach has a tendency not to build models that over-fit the data and is therefore suitable for learning a many-to-many bilingual alignment model. In [4] the model was used to align the data to be mined, and features from this alignment were used to classify the data. The classifier was trained on a set of seed sentences that were known to be correct (supplied as part of the NEWS workshop task), and a set of negative examples that were selected from the data. Their results yielded levels of precision and recall that were comparable with the best systems in the NEWS2010 shared task, for all of the language pairs tested. One weakness of their approach is that they make no attempt to screen out noisy data when training their alignment model, and erroneous parameters learned from the noise may degrade the performance of the model for some types of data (such as the dictionary data used in these experiments). In other words, their system may learn to model the noise, and as a consequence learn to mine pairs that are similar in character to noisy examples that were trained on. This paper extends their approach to include an explicit noise model in order to mine word pairs in a low-resource environment, and describe these extensions later in the paper. The next section introduces the methods that are used to introduce human knowledge into the statistical systems.

4. PHONETIC SIMILARITY GROUPING

Every language has its unique phonetic scheme that consists of phonemes, phonetic rules, the rhythm, stress, and intonation of speech [13]. Thus, ambiguity arises when the sounds map across the phonetic systems. Usually transliteration mining systems have to face the problem of scripts in which:

1. the languages have similar phonemes but varied scripts (for example, "ဗီတာမင်"(bi.ta.min) and "Vitamin" in Myanmar-English);
2. the languages have different phonemes and similar scripts (for example, "加速度"(Kasokudo) in Japanese and "加速度"(Jiāsùdù) in Chinese);
3. the languages have similar phonemes and similar scripts (for example, "атом"(atom) and "ATOM" in Russian-English);
4. the languages have different phonemes and different scripts (for example, "加速度" (Jiāsùdù) and "acceleration" in Chinese-English).

However, the aim of the current research is to deal with the extraction of phonetically similar transliterated word pairs (i.e., case 1 in the above list) from parallel or comparable corpora automatically.

Phonetic similarity is used to compare two data strings that may be spelled differently but sound the same. Soundex [14] is a phonetic algorithm for indexing names by their sound when pronounced in English that has found widespread application in the linguistics domain over a long period of time in the indexing system of database (e.g. U.S. National Archives), and is still finding application in recent years for example in Oracle[2], MYSQL[3], Microsoft SQL[4] Server etc.. In principle, Soundex is used to convert English words by simplifying the phonetic representation based on the six acoustic categories shown in Table 1.

Table 1. Soundex Code System (U.S. English)

Code	Letters	Description
1	B, F, P, V	Labials
2	C, G, J, K, Q, S, X, Z	Gutturals & Sibilants
3	D, T	Dental
4	L	Long Liquid
5	M, N	Nasal
6	R	Short Liquid
Delete	A, E, H, I, O, U, W, Y	Vowels plus H, W, & Y

In the Soundex representation, names with the same pronunciation is encoded to the same string, therefore matching can occur despite minor differences in spelling like "Smith" and "Smythe". The original Soundex codes have four alphanumeric characters consisting of a letter and three numbers. The letter is always the first letter of the surname and then the numbers are assigned to the remaining letters of the surname. All vowels are dropped except those occurring in the first letter, because it is based on the phonological concept that vowels are not pronounced.

When assigning codes and phonetic categories to languages other than English, typically the characters cannot be directly mapped onto the original Soundex categories. Therefore, Soundex is usually adapted for use with other languages by taking into account the language's specific characteristics.

In these experiments, both English and Myanmar scripts are transcribed into a phonetic coding and then categorized into phonetic similarity groups. The human knowledge-based similarity metric is based on a phonologically motivated model of similar sound groups (based on Soundex and the use of the International Phonetic Alphabet-IPA to represent the phonetic transcription) [15]. The IPA can represent phonetic transcription of speech sounds for all languages, but this research does not require such distinctions. Thus, some trivial distinctions are grouped together and some different phonetic symbols (i.e., IPA symbols) are simplified into one Latin letter to simplify the sound in different languages. The phonetic grouping is dependent on the articulation position and manner of sound, and each group is assigned a unique phonetic code [16]. For example, although the letters (C, G, J, K, Q, S, X, Z) are coded in a group in Soundex, this mapping treats different groups separately based on the articulatory manner of plosive and fricative. Moreover, as observed this phonetic mapping system is applicable to other languages based on their specific phonemes based on evidence from experiments in cross-language phonetic similarity in eight different languages (English, Japanese, Korean, Malay, Thai, Myanmar, Indonesian, and Vietnamese) [17]. Table 2 specifies the relation of phonetic features, phonetic symbols, baseline IPAs, and the assigned codes used in this experiment.

[2] http://docs.oracle.com/cd/B19306_01/server.102/b14200/functions148.htm

[3] http://dev.mysql.com/doc/refman/5.0/en/string-functions.html#function_soundex

[4] http://msdn.microsoft.com/en-us/library/aa259235%28v=sql.80%29.aspx

Table 2. Phonetic Similarity Grouping

Articulatory Place	Articulatory Manner	Symbols	IPA	Assigned Code	Articulatory Place	Articulatory Manner	Symbols	IPA	Assigned Code
Labial	Plosive	p	p	1	Velar	Plosive	k, kh	k	6
		b	b				c		
		V, ph	ʋ				q	q	
	Fricative	f	f	2			x	χ	
		v	v				g	g	
Dental	Fricative	th	θ	3		Fricative	X	ç, x, (χ)	7
		dh	ð				h	h	
		t	t (t̪)				H	ɦ, ħ, ʕ	
		d	d (d̪)				Q	ɣ, ɟ, ʙ	
Alveolar	Plosive	j, ge	ɟ, dʒ	4	r-Sound		r	ʀ	8
	Fricative	s	s	5			rr	ɾ, ʁ, ɹ, ɻ, ʀ, ɽ, ɺ	
		z	z		l-Sound		l	ʟ	9
		sh	ʃ				LY	ɬ, ɫ, ʟ, ɭ, ɮ	
		zh	ʒ		Approximants		y	j	B
		ch	tʃ				w	w	
Nasals		m, hm	m	A			yy	ɥ	
		my	ɱɣ				hw	ʍ	
		n	n		Initial Vowel Sounds		a, e, i, o, u	ʔ	C
		N, hn	ɳ		Vowel		a, e, i, o, u	a, æ	ignore/delete
		ng	ŋ						
		ny	ɲ						

5. METHODOLOGY

The proposed method allows a human to provide a structure for the edit costs that are based around the phonetically-motivated model of phoneme sound groups described in the previous section. The machine is able to determine precise values for these costs within three different frameworks: a stochastic string-edit distance [3], stochastic sting-edit distance with noise model [11] and a Bayesian alignment approach [4, 12]. Figure 1 presents an overview of the experimental procedure used to evaluate the effectiveness of these frameworks which are described in the following sections..

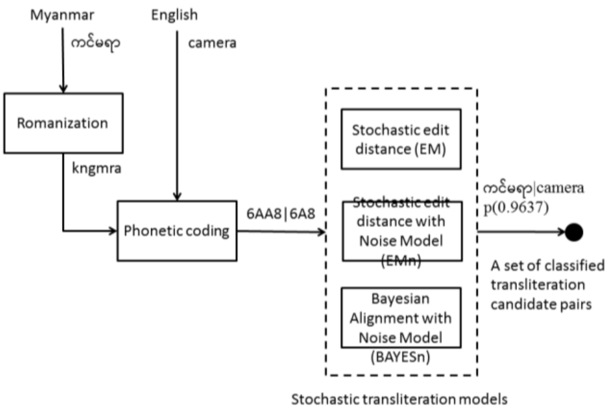
Myanmar
English
ကင်မရာ
camera
Romanization
kngmra
Phonetic coding
6AA8|6A8
Stochastic edit distance (EM)
Stochastic edit distance with Noise Model (EMn)
Bayesian Alignment with Noise Model (BAYESn)
ကင်မရာ|camera p(0.9637)
A set of classified transliteration candidate pairs
Stochastic transliteration models

Fig 2: Experimental framework

5.1 Stochastic edit distance

String Edit Distance (Levenshtein distance) [10] measures the dissimilarity between strings and is defined as the minimum number of edit operations needed to transform one string into the other, where an edit operation is an insertion, deletion, or substitution. Identity substitutions are defined to have zero cost. The edit distance can be calculated by a simple edit operation count, or each edit operation can be assigned its own cost. Often these edit costs are assigned plausible values by hand. As an example, a list of edit operations and their associated edit costs from a string which is represented as the sequence X=(a,b) to the string Y=(c,b) are given below:

$$\text{cost}((a,\phi)) = 1 \text{ (deletion)}$$

$$\text{cost}((\phi,c)) = 1 \text{ (insertion)}$$

$$\text{cost}((a,c)) = 1 \text{ (substitution)}$$

$$\text{cost}((b,b)) = 0 \text{ (identity)}$$

Using the above edit operations and costs, the Levenshtein distance is given by the minimum number of edits. In this example, only a single edit is required - a substitution of c for a - and the Levenshtein distance is therefore 1.

The standard Levenshtein distance for string sequences was given a stochastic interpretation by [3], where a stochastic transducer was used to define two string edit distances: the Viterbi edit distance and the stochastic edit distance. The model is described in some detail below, as it forms the basis for the models in this research paper.

In a stochastic framework, the generation process is governed by a generative model that can assign a joint probability to a pair of strings using probabilities on edit operations. Under their interpretation, the joint probability is transformed into an edit cost by taking the negative logarithm. A given example illustrates visually how a stochastic edit distance is derived from the joint probability of strings $X=(a,b)$ and $Y=(c,b)$ generated by a memoryless stochastic transducer (all possible paths are shown in figure 2).

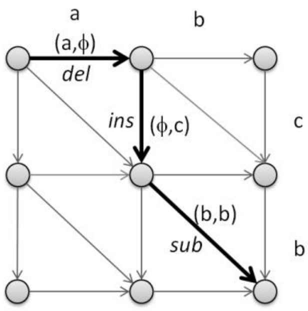

Fig 2: An edit path

In the above figure, horizontal arcs represent deletions, vertical arcs represent insertions and diagonal arcs represent identity/substitution operations. One of the edit paths is shown in bold and the corresponding sequence of edit operations is listed below:

(a, ϕ) = deletion

(ϕ, *c*) = insertion

(*b*, *b*) = identity/substitution

The probability of the cost of a single path *s* is the product of each associated edit cost belonging to the path and generally defined as:

$$P(s) = \prod_{e\epsilon s} P(e)$$

where *e* is an edit, and s $=(e_1, e_2, e_3, ...e_n)$.is a sequence of edits (an edit path).

A source string *X* and a target string *Y* can be aligned in many ways, corresponding to multiple paths in the graph. The Viterbi edit distance is defined using the most likely edit sequence for the string pair <*X*,*Y*>.

$$P_v(X,Y) = max_{s\epsilon Z} P(s)$$

where $Z = \{s_1, s_2, s_3,....,s_j\}$ is the set of all edit operation sequences that can generate *X* and *Y*. The stochastic edit distance $d_s(X,Y)$ is defined as the negative logarithm of the joint probability of the string pair $P(X,Y)$ according to a memoryless stochastic transducer [3, 11]. This is calculated by summing the derivation probabilities over all paths in the graph in Figure 2.

$$d_s(X,Y) = \sum_{s\epsilon Z} \sum_{e\epsilon s} -log(P(e))$$

The parameters of the memoryless stochastic transducer (which will be reinterpreted as the edit costs) are learned with the Expectation Maximization (EM) algorithm using a forward-backward dynamic programming technique for efficiency. The EM algorithm finds a locally optimal set of parameters in terms of the likelihood of the model given the data.

5.2 Stochastic Edit Distance with Noise Model

When applied to transliteration mining, the stochastic model of edit distance described in the previous section can be extended by adding a noise model (a non-transliteration sub-model) to the transliteration sub-model [11]. The full transliteration mining model being an interpolation of both models:

$$P(X,Y) = (1-\lambda)P_t(X,Y) + \lambda P_n(X,Y)$$

Where λ is the prior probability of the data being noise (a non-transliteration pair), P_t is the probability of transliteration sub-model, and P_n is the probability of noise model.

The transliteration sub-model aligns the characters to each other or to null using a unigram model. The EM training of the transliteration model is similar to that process described in the previous section. The noise model randomly generates characters using two independent unigram models, and is estimated once at the start of training from the training data. After EM training, the transliteration word pairs are expected to be assigned a high probability by the transliteration sub-model and a low probability by the noise model. This model will be referred as EMn (EM alignment with noise model).

5.3 Bayesian Alignment with Noise Model

This model incorporates the essence of the ideas proposed in [11] into a non-parametric Bayesian learning framework. It contains a similar explicit noise model, and to do so introduces an additional generative step that selects the type of word pair the model will generate. This technique will be referred to as BAYESn (Bayesian alignment with noise model). The structure and characteristic of this model is described as follow:

5.3.1 Overfitting

The motivation for extending the EM model to a Bayesian model is the desire to use many-to-many alignment. One-to-one alignment is useful when aligning two sequences of romanized characters, but usually cannot be used for non-roman scripts. A major limitation of maximum likelihood training when applied to bilingual alignment is its tendency to overfit the data. Assigning a large amount of probability mass to long sequence pairs in the data will produce a model with a high likelihood. In the most extreme case where there are no restrictions on the source and target sequence lengths in the many-to-many mapping, the most likely model will assign a probability of one to a single alignment of the entire source side of the corpus to the entire target side. Nonparametric Bayesian models discourage the addition of long pairs into the model, by assigning them a low probability and by rewarding the re-use of parameters in their models [2].

5.3.2 Model Structure

Figure 2 shows the structure of the model which consists of two square graphs corresponding to the transliteration sub-model and the noise sub-model. The generative story for the model is a 2-step process as follows:

Step 1 : Choose whether to generate a noise pair (with probability λ) or a transliteration pair (with probability 1-λ);

Step 2 : Generate a pair of the chosen type using the appropriate model.

For details of how the transliteration sub-model is trained the reader is referred to [4]. This model differs from theirs in that it performs clustering into two classes as it learns the probabilities for its model parameters. The transliteration sub-model is thereby trained using only those clean samples that fall into the transliteration class. The λ probability is updated based on a simple frequency count whenever a transliteration candidate is assigned a new class. It is possible either to train the noise model in the same manner as [11] so that it is trained only on noisy data, or it can be trained once at the start of training from the whole of the training corpus. According to the observation of pilot experiments, both of these techniques are effective and gave approximately equal performance. The model is chosen to train once at the start of the training for the experiments in this paper. The classes for the candidate pairs were assigned randomly in the first iteration of training using a noise probability λ=0.5.

In [12], the aligner performed a forced many-to-many alignment in the spirit of the alpha/beta edit operations proposed in [18], but did not include the capability to make null alignments. In this work their model is extended to allow null alignments to multiple characters in both languages. In Figure 2 for simplicity only one-to-one alignment is illustrated, but in reality arcs that traverse greater distances in the graph are possible but are limited by parameters that control the maximum spans of an edit operation in terms of the number of source and target characters it can operate on.

As shown in the figure below, a generative model generates *X* and *Y* clustering and aligning into two sub-models: transliteration model and noise model. An edit path of each sub-model is shown in bold and the corresponding sequence of edit operations is listed on the right.

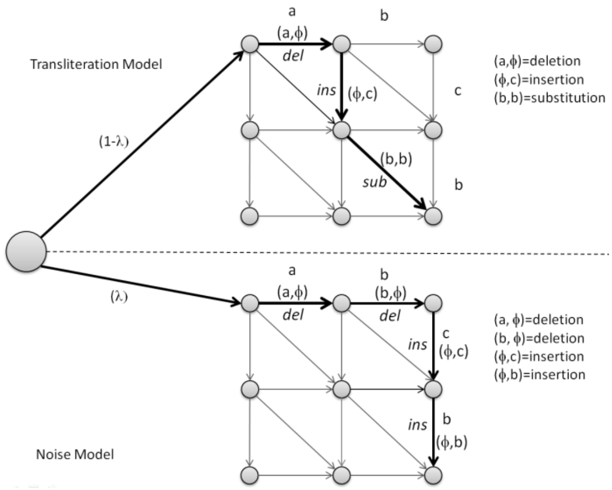

Fig 2: Sub-models of BAYESn

6. EXPERIMENTS

6.1 Experimental Data

For these experiments a Myanmar-English bilingual training corpus consisting of 1729 word pairs from the titles of Wikipedia articles linked by inter-language links⑤was used. The corpus additionally included 11462 word pairs extracted from a Multilingual Terminology Dictionary⑥. From this complete data set, two derivative training sets are constructed. The first consisted of 3100 single word pairs, which is called 'clean data'; this type of data is typical of the kind of data one might encounter in a shared task on transliteration mining such as the NEWS workshop [19]. The second consisted of 14891 multiple word pairs in which a pair may consist of more than one word in English; this data is called 'realistic data' as for this task, where resources are low, as observed it is necessary to mine data in a less than a ideal format.

6.1.1 Data Annotation

The testing data were manually annotated as transliteration and non-transliteration pairs by a bilingual human annotator. Of the 3100 bilingual pairs sampled, 1291 pairs were transliterations and 1809 pairs were non-transliterations.

6.2 Pre-processing

6.2.1 Data Cleaning

The source data (Myanmar) contained a lot of noise such as spelling errors and issues with mixed encoding. Moreover, there are many technical problems in Myanmar Unicode characters, for example, the case of U+200C (Zero-width non-joiner) is a non-printing character used in the writing system, that needs to be eliminated. Some other usages like "။", "၊", white spaces, and the Unicode born were also eliminated.

6.2.2 Segmentation

The Myanmar language is syllable-timed; therefore a preprocessing task was required to break it into syllables (syllable segmentation). As an example, according to the Unicode⑦ encoding standard, the syllable ('ကျောင်း') is encoded as follows:

က + ◌ျ + ေ◌ + ◌ာ + င + ◌် + ◌း = ကျောင်း

The syllable ('ကျောင်း') basically consists of an initial consonant ('က') with optional medials ('◌ျ'), dependent vowels ('ေ◌','◌ာ'), dependent signs ('◌း'), and more than one consonant may appear together with the devoweliser (the killer character Asat '◌်').

A segmentation process consisting of 3 rules is necessary to segment the Myanmar grapheme sequences into sequence of syllables (this experiment used a tool that is developed by Ye Kyaw Thu, NICT (2012)).

Rule-1: Break in front of consonant, independent vowel, number and symbol characters" and is the first step for syllable breaking. But there is an exception for Kinzi ('င်္◌'), i.e., a combination of a conjunct (U+1004) with Myanmar letter ('င') preceding the consonant. For example, the Myanmar word "ကွန်ပျူတာ" can be segmented into syllables as "ကွန်|ပျူ|တာ".

Rule-2: Remove the breaking point in front of a subscript consonant (i.e., PadSint). For example, "မိတ္တူ" breaks as |မိ|တ္|တူ| and replaces with Asat |တ်| and finally combines with a front segmented letter "မိတ်|တူ".

Rule-3: Break in front of Kinzi character ('င်္◌'). For example, "အင်္ဂလိပ်" breaks as "အင်|ဂ|လိပ်".

6.2.3 Romanization

Furthermore, the Myanmar script is non-alphabetic, and therefore there are two different romanization schemes to convert Myanmar into the Latin alphabet in order to study the

⑤ http://dumps.wikimedia.org/mywiki/20120824/

⑥ http://gii2.nagaokaut.ac.jp/mtd/

⑦ http://www.unicode.org/standard/standard.html effect of the using differing romanization schemes on mining performance. The romanization systems used in this experiment were: the Myanmar Language Commission (MLC) transcription system [20] and the University of Foreign Language (UFL) pronunciation system [21] that is an extended version of MLC, but is significantly different in character. Table 3 shows three examples of the two romanization schemes in use. Romanized words in UFL appear to be more similar to the spelling of the word in English.

Table 3. Romanization schemes

English	Myanmar	MLC	UFL
vitamin	ဗီတာမင်	bitamang	bi.ta.min
motorcar	မော်တော်ကား	mautauka:	mo.to.ka
platinum	ပလက်တီနမ်	paktinam	ple'ti.nam

6.2.4 Phonetic Coding

Training data in both languages were mapped into the phonetic code strings using this technique for phonetic similarity grouping described in Section 4. Table 4 shows three word pairs transcribed into phonetic code sequences. Note that although the spellings of each of the romanized forms differ significantly, the phonetic coding sequences are all identical.

Table 4. Romanization schemes

	English	MLC	UFL
Romanization	vitamin	bitamang	bitamin
Phonetic coding	1030A0A	1030A0A	1030A0A

6.3 Evaluation Criteria

In order to evaluate the performance of these models, the precision, recall, and f-score evaluation metrics are used, which are described as follows: precision, recall, and f-score. Where TP is the number of correct pairs (transliteration pairs) that were labeled as correct (true positive), FP is the number of incorrect pairs (non-transliteration pairs) that were labeled as correct (false positive), and FN is the number of correct pairs (transliteration pairs) that were labeled as incorrect (false positive). When mining real data, the data may not necessarily be mined at the optical F-score; an appropriate trade-off between precision and recall may need to be selected to fit the specific application. For this reason the results are presented in the form of graphs of the complete precision/recall curves (shown in Figures 3-8). In addition a summary of the results is provided at optimal F-score in Tables 5 and 6 for both UFL and MLCTS Romanization.

6.4 Training

6.4.1 EM Training

The EM model was trained on the 3100 single words of the clean data set that were both romanized and phonetically coded. In order to eliminate the noise model, the EMn model is trained by fixing the parameter λ (the probability of non-transliteration) at zero. It was found that EM model could not learn a good model from both MLCTS and UFL training data, both of which gave poor mining performance (See Figures 3 and 4).

6.4.2 EMn Training

The EMn model showed improved performance on phonetically coded data over the models trained on romanized data (see Figure 5). The prior probability of λ was initially set to a value between 0 and 1. The results showed that performance was dependent on the romanization scheme used, and that the results in UFL were better than MLCTS data (See Figures 5 and 6).

Table 5. Mining performance using UFL romanization

	Data type	Precision	Recall	F-measure
EM	Romanization	0.4241	0.8574	0.5675
	Phonetic coding	0.4236	0.8466	0.5647
EMn	Romanization	0.9601	0.9147	0.9369
	Phonetic coding	0.9825	0.9132	0.9466
BAYESn	Romanization	0.9642	0.9186	0.9408
	Phonetic coding	0.9785	0.955	0.9666

Table 6. Mining performance using MLCTS romanization

	Data type	Precision	Recall	F-measure
EM	Romanization	0.4637	0.7327	0.5680
	Phonetic coding	0.4364	0.7606	0.5546
EMn	Romanization	0.4419	0.9140	0.5958
	Phonetic coding	0.9166	0.8946	0.9055
BAYESn	Romanization	0.9555	0.9326	0.9439
	Phonetic coding	0.9706	0.9473	0.9588

6.4.3 BAYESn Training (Clean Data)

The BAYESn model learned models with comparable performance to the EMn model on the clean data set for both romanized and phonetically coded data. The results again showed that using the phonetic coding gave a much better performance. The results are shown in Figures 7 and 8.

6.4.4 BAYESn Training (Realistic Data)

The results from the BAYESn model trained on the 14891-sample realistic dataset show similar characteristics to the results from clean data: a good-performing model is learned in both cases, but the result in higher performance is shown from the phonetically coded data with respect to the romanized data. The results are shown in Figure 9.

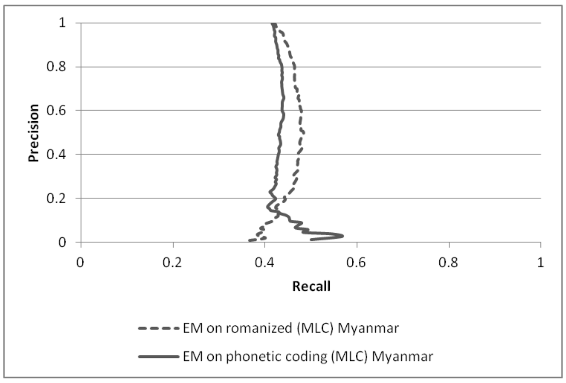

Fig: 3 Comparing MLCTS romanization to phonetic coding with the EM model

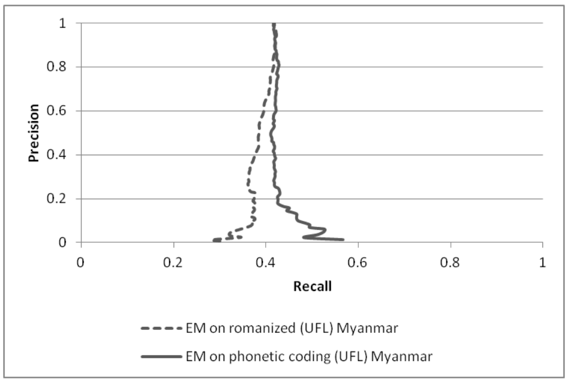

Fig: 4 Comparing UFL romanization to phonetic coding with the EM model

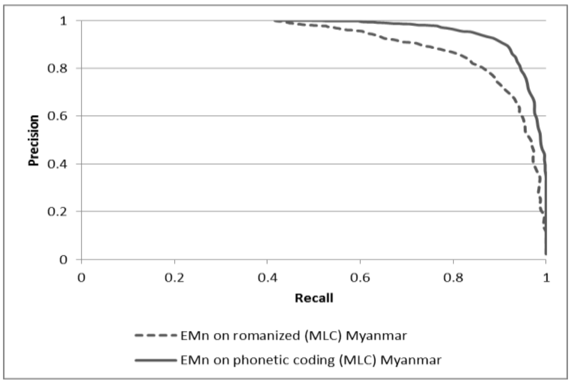

Fig: 5 Comparing MLCTS romanization to phonetic coding with the EMn model

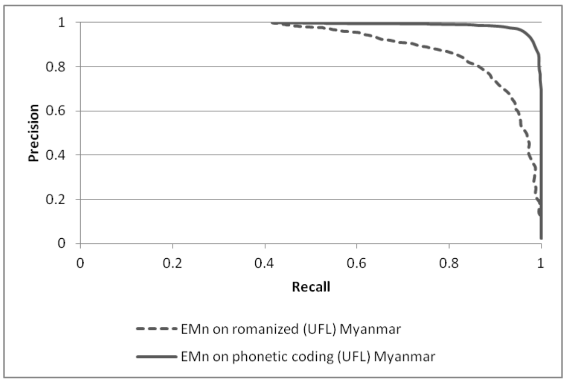

Fig: 6 Comparing UFL romanization to phonetic coding with the EMn model

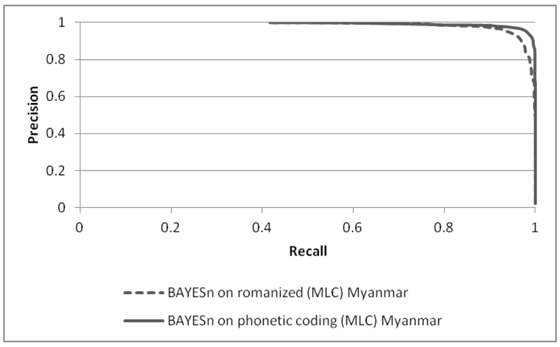

Fig: 7 Comparing MLCTS romanization to phonetic coding with the BAYESn model

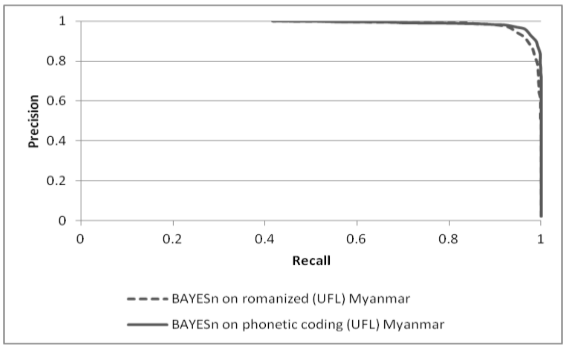

Fig: 8 Comparing UFL romanization to phonetic coding with the BAYESn model

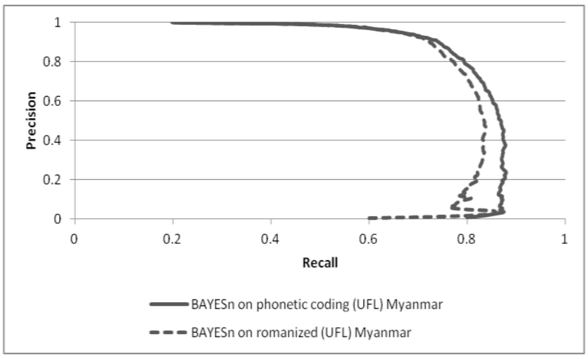

Fig 9: Comparing UFL romanization to phonetic coding with the BAYESn model using realistic data

Table 7 presents the results of BAYESn experiment using UFL Romanization and many-to-many alignment. Though phonetic group knowledge helps to learn to obtain better precision and recall on this dataset; the precision will discuss this in detail in the next section.

Table 7. The performance of BAYESn in multiple word data

	Data type	Precision	Recall
BAYESn	Romanization	0.79	0.72
	Phonetic coding	0.81	0.74

6.4.5 Learning the Noise Prior

The EMn model is trained by using an initial value for λ (the prior probability of the candidate pair being noise) of 0.6. After training had been completed the model arrived at a value of 0.6152 for lambda. The value of lambda is estimated to be 0.5835 from human-assigned labels of all of the 3100 pair clean data set. The value learned during the EMn training is commensurate with the true value.

The BAYESn model learns a similar parameter λ, which also represents the prior probability of the candidate pair being noise. This model can not only provide an estimate for the parameter, but also a distribution indicating its uncertainty. The convergence of the value of the parameter during training is shown in Figure 10. It can be seen from the graph that the value of this parameter converges after about 10 training iterations. The training is continued until iteration 50, and then sampled its value over the next 500 iterations. A histogram of representing the distribution of the value of λ for this 500-iteration sample is shown in Figure 11. This distribution has a mean of 0.580 which is extremely close to the ground truth value of 0.5835.

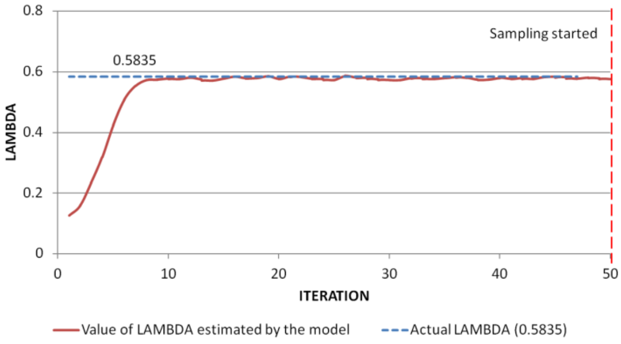

Fig: 10 Convergence of the noise prior (λ)

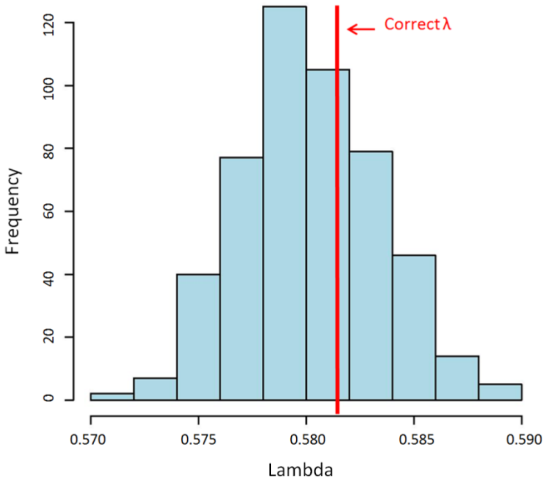

Fig: 11 The distribution of λ from 500 samples taken after training completion

6.4.6 Using Different Romanization Systems

There are a variety of systems of romanizing Myanmar in use today. Some systems place emphasis on the orthography of the Myanmar script (transliteration), and other focus on the pronunciation of the Myanmar words (transcriptions) [22]. The experiments used the Myanmar Language Commission (MLC) Transcription System and the University of Foreign Language (UFL) pronunciation system.

The results (presented in Figures 5, 6, 7 and 8) consistently show that for phonetic similarity comparison, the choice of romanization system matters, and in the case of these experiments, the UFL scheme always gave better results than using the MLC system. The UFL system is primarily intended for use by foreigners to pronounce the Myanmar language and this should make it more similar to the English than the MLC system which, as just one example, includes characters in the romanized form that represent non-articulated consonants.

To illustrate this point, consider the differences in how the word "platinum" is romanized by the two systems. In MLC it is romanized as "plaktinam" whereas in UFL it is romanized as "ple'ti.nan". Phonetically coding the ULF romanized form gives rise to the same coding as the original English coding (See in Tale 8).

Table 8. An example difference between MLC and UFL

	English	**MLC**	**UFL**
Romanization	platinum	pla**k**tinam	ple'ti.nam
Phonetic Coding	19030A0A	190**6**30A0A	19030A0A

Moreover, the nature of Myanmar language is syllable-timed, whereas English is stress-timed, and consequently Myanmar words do not have final consonant [22]. In the example above for the word platinum, written " ပလက်တီနမ်", the syllable "လက်" has a consonant "က" at the end, the "**k**" stands for "က" in the romanization but is not articulated. In pronunciation, it is easy to distinguish between a consonant and a non-articulated consonant, whereas in the transcription from Myanmar syllables to the Latin alphabet, how these consonants are represented will depend on the romanization scheme.

6.4.7 Using a Noise Model

Adding an explicit noise model to the EM and BAYES models substantially improved their mining performance on both clean data and realistic data in these experiments (see the results in Figures 5, 6, 7, 8, and 9). Because this is at least in part due to the character of the data which were used. The experimental data included word pairs extracted from a bilingual dictionary. This type of data contains multiple forms derived from the same root in which the prefixes of the words in both source and target language are identical across the word forms. An example a set of word forms and their translations is given below:

a. (စုပ်ယူနိုင်စွမ်းရည်|absorbance)

b. (စုပ်ယူနိုင်စွမ်းရည်|absorbing)

c. (စုပ်ယူအားတိုင်း|absorbing)

This set is composed entirely of non-transliterations, yet the prefixes of both source and target words are identical for all members of the set (i.e., စုပ်ယူ|absorb). The bilingual alignment models may learn erroneous features from this data since these examples will support each other when it comes to the alignments of their prefixes. Even though there is little or no support from the rest of the corpus, the mutual support from the members of the set may be enough to cause these pairs to be assigned a high enough probability to be mined as correct pairs. Using noise model can be reduced this type of problem, but it did not eliminate all from the alignments.

6.4.8 Using Phonetic Grouping

Using this proposed method for phonetic grouping gave the best overall mining performance in the experiments and at the same time speeded up the learning tasks. The processing time for each method with and without phonetic grouping is measured and found that using phonetic grouping can reduce the execution time by about 60%~70%. The experiments measured 200-iterations of training on the clean dataset, and the results are shown in Table 9.

Table 9. Execution Time

Model	Data	Execution Time (minutes)
EM	MLC with phonetic coding	7.13
	MLC without phonetic coding	25.54
	UFL with phonetic coding	4.86
	UFL without phonetic coding	16.17
EMn	MLC with phonetic coding	8.77
	MLC without phonetic coding	27.93
	UFL with phonetic coding	5.04
	UFL without phonetic coding	16.42

As observed in all experiments, performance in transliteration mining was increased by using phonetic grouping (see in Figures 5 & 6). The EMn method gave improved precision using phonetic grouping compared to the results obtained by using romanization. For example, EMn erroneously labeled the bilingual pair (2011|nhshtnghsyti)[8] as a true transliteration pair with probability 0.993203, whereas EMn correctly rejected this pair (the probability of it being a transliteration was 0.001139) when phonetic grouping was employed. This is due to the fact that the phonetic grouping approach improves to the performance of model.

From the experimental results with clean data, it was observed that the performance of the approach on Myanmar data using phonetic grouping was sufficiently high to make this technique useful in a real-world mining application: the BAYESn approach model achieved scores of 0.97 in precision, 0.95 in recall, and 0.96 in F-measure. In addition, when mining with realistic data, the results showed that not only did BAYESn model learn a strongly performing model within the human expert knowledge framework, but it also speeded up the learning procedure.

6.4.9 Aligning without Romanization

In this experiment a task was aligned directly from Myanmar syllables to English characters in a many-to-many manner. This has the advantage of removing any requirement for a romanization system, thereby making it far more generally applicable.

A dataset containing 13,483 multiple word pairs without Romanization was used in these experiments, and before aligning the data it is first segmented into syllables using the procedure described in Section 6.2.2.

In the many-to-many alignment procedure it is possible to constrain the maximum source and target character sequence sizes. The experiments were run in 3 different settings of these parameters to investigate their effect, constraining both source and target sequences to be of length 4, 8 and 12 tokens. The results are shown in Table 10, and indicate that the model tends to improve with fewer restrictions on the sequence lengths. This is consummate with the hypothesis that the model does not have a tendency to overfit the data.

Table 10. Number of model parameters and accuracy

Parameter	Model Parameters	Accuracy
4-4	9927	69.93%
8-8	11625	76.28%
12-12	11717	77.58%

Although the results achieved a somewhat satisfactory level, the performance was considerably lower than the systems that used romanized data. As observed the result showed that this is due to the nature of the Myanmar language itself. There are around 1880 unique syllables in the language, some much rarer than others, and given the small size of the data sets available for Myanmar. Therefore the amount of available data was simply insufficient to train a good alignment model given the number of parameters in the model. Visual inspection of the alignment data showed that some of the rarer syllables had very little or no data to train from. Romanization is an effective way to overcome this problem since the vocabulary size can be vastly reduced, and even the rarer syllables can be represented accurately. Phonetic grouping over the romanization is for the same reason even more effective when only small amounts of data are available, as the experiments have shown.

7. CONCLUSION AND FUTURE WORK

In this paper, a novel technique is proposed for integrating human knowledge into stochastic models of string similarity. This approach is expected to be most useful when working with low resource languages as the human knowledge can be used to mitigate issues of data sparseness resulting from lack of data; the number of edit costs that need to be learned can cause problems in low resource languages where there may not be enough data available to learn them accurately. Therefore, the proposed approach uses human expert knowledge (phonetic group coding) to provide a framework within which a machine can learn the edit costs by effectively tying parameters in a linguistically-motivated manner in order to reduce the number of parameters to be learned, thereby simplifying and speeding up the learning task. The experiments show that in a Myanmar-English transliteration mining task, this approach substantially improves mining performance when mining realistic data. In order to study the utility of performing alignment without romanization, an explicit noise model is added to a non-parametric Bayesian alignment model. The results show that this model works as well as the model based on EM training, but when applied to un-romanized data the model was not able to perform at the same level as the systems based on romanized data. As observed this approach failed for the same reason the phonetic coding approach succeeded: the direct alignment of Myanmar to English introduces too many parameters to be learned from the small amount of available data. This technique still may be viable for languages with more data, and/or smaller input grapheme/syllable set sizes where the data sparseness issues are less severe, but this remains future work.

[8] နှစ်ထောင့်ဆယ့်တစ် (Myanmar)

The experimental results clearly show that the choice of edit cost is a strong factor in determining the performance of the edit-distance-based techniques used in these experiments. Often edit costs are selected to have plausible values by human experts, but better results can be obtained through the application of machine learning techniques to learn appropriate edit costs. Furthermore, the mining performance using stochastic models depends heavily on the romanization system used, and this motivates further research in the area of string representation. All the experiments in the paper were performed using an unsupervised mining approach, and in future research it would be interesting to study realistic/noisy data mining within a semi supervised framework.

8. ACKNOWLEDGMENTS

Our thanks to KANOU Mitsuru (Associate Professor, Department of General Education/Japanese Studies, Nagaoka University of Technology) who suggested towards development of phonetic similarity grouping and Ye Kyaw Thu (NICT-Kyoto) who contributed a tool for Myanmar syllable segmentation.

9. REFERENCES

[1] Kevin Knight and Jonathan Graeh: Machine Transliteration, Journal of Association for Computational Linguistics, vol. 24, no. 4, (1998).

[2] Andrew Finch, Keiji Yasuda, Hideo Okuma, Eiichiro Sumita, and Satoshi Nakamura: A Bayesian Model of Transliteration and Its Human Evaluation When Integrated into a Machine Translation System, IEICE Transactions on Information and Systems E94-D, 10, 1889-1900, (2011).

[3] Eric Sven Ristad and Peter N. Yianilos: Learning String-Edit Distance, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 5, (1998).

[4] Takaaki Fukunishi, Andrew Finch, Seiichi Yamamoto, Eiichiro Sumita: A Bayesian Alignment Approach to Transliteration Mining, ACM Transactions on Asian Language Information Processing, vol. 9, no. 4, article. 39, (2012).

[5] K Saravanan, Raghavendra Udupa and A Kumaran: Improving Cross-Language Information Retrieval by Transliteration Mining and Generation, proceedings of Tamil Internet Conference, in Philadelphia, (2011).

[6] He, X: Using word dependent transition models in HMM based word alignment for statistical machine translation, proceeding of 2nd ACL Workshop on Statistical Machine Translation, (2007).

[7] Kareem Darwish: Transliteration Mining with Phonetic Conflation and Iterative Training, proceedings of the 2010 Named Entities Workshop, ACL 2010, pages 53-56, (2010)

[8] Ali EI Kahki, Kareem Darwish, Ahmed Saad EI Din, Mohamed Abd EI-Wahab, Ahmed Hefny, and Waleed Ammar: Improved Transliteration Mining Using Graph Reinforcement, proceedings of EMNLP '11 Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 1384-1393, (2011).

[9] Ali EI Kahki, Kareem Darwish, Ahmed Saad EI Din, and Mohamed Abd EI-Wahab: Transliteration Mining Using Large Training Test Sets, proceedings of 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 243-252, (2012).

[10] V. I. Levenshtein: Binary codes capable of correcting deletions, insertions, and reversals, Journal of Soviet Physics Doklady, vol. 10, no. 8, pp.707–709, (1966).

[11] Hassan Sajjad, Alexander Fraser, and Helmut Schmid: A Statistical Model for Unsupervised and Semi-Supervised Transliteration Mining, proceedings of Association for Computational Linguistics (ACL-2012) conference, (2012).

[12] Andrew Finch and Eiichiro Sumita: A Bayesian Model of Bilingual Segmentation for Transliteration, proceedings of the 7th International Workshop on Spoken Language Translation, pages 259-266, (2010).

[13] Jin-Shea Kuo, Haizhou Li, and Ying-Kuei Yang: A Phonetic Similarity Model for Automatic Extraction of Transliteration Pairs, ACM Trans. Asian Language Information Processing, vol. 6, no.2, article 6, (2007).

[14] R. C. Russell and K. M. Odell: Soundex phonetic comparison system [cf. U.S. Patents 1261167(1918), 1435663 (1922)], USA, (1922).

[15] David Odden: Introducing Phonology, Cambridge University Press, pp.34-39, (2005).

[16] Shigeaki Kodama: String Edit Distance for Computing Phonological Similarity between Words, proceedings of International Symposium on Global Multidisciplinary Engineering, (2010)..

[17] Ohnmar Htun, Shigeaki, Kodama, Yoshiki Mikami: Cross-Language Phonetic Similarity Measure on Terms Appeared in Asian Language, International Journal of Intelligent Information Processing, vol. 2, no. 2, (2011).

[18] Eric Brill, Gary Kacmarcik, Chris Brockett: Automatically Harvesting Katakana-English Term Pairs fromSearch, Asia Federation of Natural Language Processing, (2001).

[19] A Kumaran, Mitesh Khapra, and Haizhou Li: Whitepaper on NEWS 2010 Shared Task on Transliteration Mining, Whitepaper of NEWS 2010 Shared Task on Transliteration Generation, (2010).

[20] Word: Myanmar Language Commission (MLC), http://en.wikipedia.org/wiki/MLC_Transcription_System

[21] University of Foreign Language, Yangon, Myanmar: An introductory course in Myanmar language, (2005).

[22] Myanmar Language Commission, Ministry of Education, Myanmar: Myanmar-English Dictionary, 9th Edition, (2008)