Machine Transliteration Survey

SARVNAZ KARIMI
NICTA and The University of Melbourne
and
FALK SCHOLER
RMIT University
and
ANDREW TURPIN
RMIT University

Machine transliteration is the process of automatically transforming the script of a word from a source language to a target language, while preserving pronunciation. The development of algorithms specifically for machine transliteration began over a decade ago based on the phonetics of source and target languages, followed by approaches using statistical and language-specific methods. In this survey, we review the key methodologies introduced in the transliteration literature. The approaches are categorized based on the resources and algorithms used, and the effectiveness is compared.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey

General Terms: Algorithms, Experimentation, Languages.

Additional Key Words and Phrases: Automatic Translation, Machine Learning, Machine Transliteration, Natural Language Processing, Transliteration Evaluation.

1. INTRODUCTION

Machine translation (MT) is an essential component of many multilingual applications, and a highly-demanded technology in its own right. In today's global environment the main applications that require MT include cross-lingual information retrieval and cross-lingual question answering. Multilingual chat applications, talk-

Authors' addresses: S. Karimi, National ICT Australia, Victoria Research Laboratory, The University of Melbourne, Parkville, Vic 3010, Australia; email: skarimi@unimelb.edu.au; F. Scholer, School of Computer Science and Information Technology, RMIT University, Melbourne, Vic 3001, Australia; email: falk.scholer@rmit.edu.au; A. Turpin, School of Computer Science and Information Technology, RMIT University, Melbourne, Vic 3001, Australia; email: andrew.turpin@rmit.edu.au.
Corresponding author: Sarvnaz Karimi, skarimi@unimelb.edu.au.
This work is based on the first author's PhD thesis completed at RMIT University, Melbourne, Australia.

ing translators, and real-time translation of emails and websites are some examples of the modern commercial applications of machine translation.

Conventionally, dictionaries have aided human translation, and have also been used for dictionary-based machine translation. While typical dictionaries contain around 50,000 to 150,000 entries, in practice, many more words can be found in texts. For example, a collection of Associated Press newswire text collected over 10 months has 44 million words comprising 300,000 distinct English words[1]. The "out-of-dictionary" terms are typically names, such as companies, people, places and products (Dale, 2007). In such cases transliteration — where the out-of-dictionary words are spelled out in the target language — is necessary.

Machine transliteration emerged around a decade ago as part of machine translation to deal with proper nouns and technical terms that are translated with preserved pronunciation. Transliteration is a sub-field of computational linguistics, and its language processing requirements make the nature of the task language-specific. Although many studies introduce statistical methods as a general-purpose solution both for translation and transliteration, many of the approaches introduced in the literature benefit from specific knowledge of the languages under consideration.

In this survey, we first introduce the key terminology and linguistic background useful for understanding the rest of the paper. A general discussion of the challenges that automated transliteration systems face, including scripts of different languages, missing sounds, transliteration variants, and language of origin follows the *Key Concepts* section. Then, the specific terminology and formulations used throughout this survey are introduced. Description of the state-of-the-art machine transliteration methods follows the formulation section. Literature on transliteration falls into two major groups: *generative transliteration* and *transliteration extraction*. Generative transliteration focuses on algorithms that transliterate newly appearing terms that do not exist in any translation lexicon. Transliteration extraction, on the other hand, enriches the translation lexicon using existing transliteration instances from large multilingual corpora such as the Web, to reduce the requirement for on-the-fly transliteration. This second category is also considered as a method of extracting large and up-to-date transliterations from live resources such as the Web. We review both of these categories, with an emphasis on generative methods, as these constitute the core of transliteration technology. We also examine the evaluation procedure undertaken in these studies, and the difficulties that arise with non-standard evaluation methodologies that are often used in the transliteration area.

2. KEY CONCEPTS

Some of the common linguistic background concepts, and general terminology used throughout this survey, are explained in this section[2]. More detailed information on writing systems, alphabets, and phonetics of different languages can be found in IPA (International Phonetic Alphabet[3]) publications, available for all the existing languages.

[1]The statistics are on words with no pre-processing such as lemmatization.
[2]Definitions are based on Crystal (2003, 2006).
[3]`http://www.omniglot.com/writing/ipa.htm`

Phonetics and Phonology. Phonetics is the study of the sounds of human speech, and is concerned with the actual properties of speech sounds, their production, audition and perception. Phonetics deals with sounds independently, rather than the contexts in which they are used in languages. Phonology, on the other hand, studies sound systems and abstract sound units, such as phonemics, distinctive features, phonotactics, and phonological rules. Phonology, therefore, is language specific, while phonetics definitions apply across languages. The phonetic representation of a sound is shown using [ ], and the phonemes are represented by / /. For example, the phonetic version of both the Persian letter "پ", and the English letter "p" is [p].

Phoneme. A phoneme is the smallest unit of speech that distinguishes meaning. Phonemes are the important units of each word, and substituting them causes the meaning of a word to change. For example, if we substitute the sound [b] with [p] in the word "big" [bɪg], the word changes to "pig". Therefore /b/ is a phoneme. Note the smallest physical segment of sound is called *phone*. In other words, phones are the physical realization of phonemes. Also, phonic variety of phonemes are called *allophones*.

Some transliteration algorithms use phonemes to break down words into their constituent parts, prior to transliteration (explained in Section 5.1, Phonetic-based transliteration systems).

Grapheme. A grapheme is the fundamental unit in written language. It includes alphabetic letters, Chinese characters, numerals, punctuation marks, and all the individual symbols of any writing system. In a phonemic orthography, a grapheme corresponds to one phoneme. In spelling systems that are non-phonemic — such as the spellings used most widely for written English — multiple graphemes may represent a single phoneme. These are called digraphs (two graphemes for a single phoneme) and trigraphs (three graphemes). For example, the word "ship" contains four graphemes (s, h, i, and p) but only three phonemes, because "sh" is a digraph.

In Section 5.2 transliteration methods that use grapheme concept are introduced (spelling-based transliteration systems).

Syllable. A syllable is a unit of pronunciation. A syllable is generally larger than a single sound and smaller than a word. Typically, a syllable is made up of a syllable peak which is often a vowel, with optional initial and final margins which are mostly consonants.

Writing system. A writing system is a symbolic system for representing expressible elements or statements in language. A writing system has four sets of specifications:

(1) a set of defined symbols that are individually called characters or graphemes, and collectively called a script;

(2) a set of rules and conventions which arbitrarily assign meaning to the graphemes, their ordering, and relations, and are understood and shared by a community;

(3) a language, whose constructions are represented and recalled by the interpretation of these elements and rules; and

(4) some physical means of distinctly representing the symbols by application to a permanent or semi-permanent medium, so that they may be interpreted.

There are four distinct writing systems called *logographic*, *syllabic*, *featural*, and *alphabetic or segmental*. Writing system of some languages fall into only one of these categories, however, some other languages use more than one of these systems.

(1) Logographic writing systems use logograms, where a single written character is used to represent a complete grammatical word. Most Chinese characters are logograms;

(2) Syllabic writing systems define a syllabary as a set of written symbols that represent or approximate syllables that constitute words. Symbols in a syllabary typically represent either a consonant sound followed by a vowel sound, or a single vowel. The Japanese writing system falls into this category;

(3) Featural writing systems contain symbols that do not represent whole phonemes, but rather the elements or features that collectively constitute the phonemes. The only prominent featural writing system is Korean Hangul. Hangul has three levels of phonological representation: featural symbols, alphabetic letters (combined features), and syllabic blocks (combined letters);

(4) Alphabetic or segmental writing systems possess an alphabet which is a small set of letters or symbols that represents a phoneme of a spoken language. The Arabic and Latin writing systems are segmental.

3. COMMON CHALLENGES IN MACHINE TRANSLITERATION

The main challenges that machine transliteration systems encounter can be divided into five categories: script specifications, missing sounds, transliteration variants, language of origin, and deciding on whether or not to translate or transliterate a name (or part of it). While other specific challenges also arise, these are less generic and generally language-pair dependant. For example in Chinese the character association in person names is gender specific (Li et al., 2007); or different impressions are conveyed based on Japanese Kanji ideograms making the selection of correct strings for a name difficult (Xu et al., 2006). We cover these other challenges on a study by study basis (Section 5 onwards).

3.1 Script Specifications

The possibility of different scripts between the source and target languages is the first hurdle that transliteration systems need to tackle. A script, as explained in Section 2, is a representation of one or more writing systems, and is composed of symbols used to represent text. All of the symbols have a common characteristic which justifies their consideration as a distinct set. One script can be used for several different languages; for example, Latin script covers all of Western Europe, and Arabic script is used for Arabic, and some non-Semitic languages written in the Arabic alphabet including Persian, Urdu, Pashto, Malay, and Balti. On the other hand, some written languages require multiple scripts, for example, Japanese

is written in at least three scripts: the Hiragana and Katakana[4] syllabaries and the Kanji ideographs imported from China. Computational processing of such different language scripts requires awareness of the symbols comprising the language; for example the ability to handle different character encodings.

While some scripts are written using separate characters (such as Latin), others introduce intermediate forms for characters that occur in the middle of a word. For example, in Persian script some letters change their form based on their position in the word. The character "پ" [p], is written "پـ" [p] when it appears at the beginning of a word, "ـپـ" [p] in the middle, and "ـپ" [p] at the end; however, this rule is sometimes violated when "پ" is adjoined to special letters such as "ا" [ɒ] in "پاپ" /pɒp/.

Another important aspect of language script is the direction in which it is written. Some languages are written right-to-left (RTL), and some are written left-to-right (LTR). For example, Persian, Arabic, Hebrew, and Taana scripts are RTL, whereas the script of English and other languages that use the Latin alphabet is LTR.

In general a transliteration system, which manipulates characters of the words, should be designed carefully to process scripts of the source and target languages, taking all of the above mentioned specifications into account. Figure 1 shows some transliteration examples in different languages with different scripts. Persian and Arabic examples are shown left-to-right at the character correspondence level to match with their English version.

In Section 5 in particular, we explain different transliteration methods that investigate different language-pairs and thus they may opt for more phonetic-based methods or orthographic methods. Particular language-pairs, may also lead to the introduction of engineering steps, such as pre-processing the data.

3.2 Missing Sounds

Different human languages have their own sound structure, and symbols of the language script correspond to these sounds. If there is a missing sound in the letters of a language, single sounds are represented using digraphs and trigraphs. For example, an English digraph "sh" corresponds to the sound [ʃ]. Cross-lingual sound translation — the function of transliteration — introduces new sounds to a target language, which the target language does not necessarily accommodate. That is, sounds cannot inevitably be pronounced the same way as in their original language after being imported to the target language. Such sounds are conventionally substituted by a sequence of sound units, which in turn are rendered to a sequence of letters in the target language. For example, the sound of [x] has no equivalent character in English and is reserved for foreign words. Many other languages support this sound, however. The equivalent Persian and Arabic letter of this sound is "خ" [x], which Persian speakers usually transliterate to the digraph "kh" in English, whereas in some other languages with Latin script, such as Czech,

[4]Katakana is a Japanese syllabary, one component of the Japanese writing system along with Hiragana, Kanji, and in some cases the Latin alphabet. The word Katakana means "fragmentary kana", as the Katakana scripts are derived from components of more complex Kanji.

Source and Tatget words	Letter Correspondence				Description
English to Persian					
John /dʒɒn/	J	o	h	n	*h* is a silent letter (no sound
	|	|		|	is associated to the letter)
جان /dʒɒn/	ج	ا		ن	and is not transliterated
Arabic to English					
نجیب /nædʒiːb/	ن	ج	ی	ب	short vowel /æ/ on N is
	|	|	|	|	normally not written in
Najib /nædʒiːb/	Na	j	i	b	Arabic script
English to Japanese					
Bill /biːl/	B	i	l	l	each syllable in Japanese is a
	\	/	\	/	consonant-vowel sequence
ビル [bi-ru]	ビ		ル		
English to Hindi					
Adam /ˈædəm/	A	d	a	m	the second "a" is not
	|	|		|	transliterated in Hindi
अदम /ˈædəm/	अ	द		म	

Fig. 1. Transliteration examples in four language pairs. Letter correspondence shows how the source and target letters aligned as they are the smallest transliteration units that correspond.

it is written as "ch". The same sound is guttural rhotic — the character "r" — in French (some accents).

Transliteration systems need to learn (usually in their training step) both the convention of writing the missing sounds in each of the languages involved in isolation, and the convention of exporting the sounds from one language to the other.

3.3 Transliteration Variants

The evaluation process of transliteration is not straightforward. Transliteration is a creative process that allows multiple variants of a source term to be valid, based on the opinions of different human transliterators. Different dialects in the same language can also lead to transliteration variants for a given source term. While gathering all possible variants for all of the words in one corpus is not feasible — simply because not all speakers of those languages can be called upon in the evaluation process — there is no particular standard for such a comparison, other than conventions developed among nations. Further, new names of companies, products, and people are introduced on a daily basis, which leaves no way of having a standard transliteration. Therefore, evaluation of transliteration systems becomes problematic, in particular when comparing the performance of different systems. We explain more on the evaluation and handling of transliteration variants when introducing evaluation metrics (Section 4.5) and our proposed evaluation scheme (Section 5.5). Other than evaluation, enriching transliteration corpora with transliteration variants is another important subject of study which is investigated in the transliteration extraction studies (Section 6).

3.4 Language of Origin

The most straightforward scenario for a transliteration system is being presented with a corpus (training and testing) that only contains instances of names in the nominated languages (that is, there is no word already transliterated or imported from other languages). However, such tidy data is rarely available. Corpora that contain source words to be transliterated from various origins are a challenge to automated systems. One challenge would be which letters to choose to represent the origin of the word. For example when transliterating the name "Josef" to Persian, one chooses the character "ژ" [ʒ] for "J" to specify this is a French name, or chooses "ج" [dʒ] for its English pronunciation. Also, the presence of words that have already been imported from a third language could lead to additional transliteration variants (Karimi, 2008; Karimi et al., 2007), or errors if not correctly specified (Huang et al., 2005; You et al., 2008). For example when transliterating the already transliterated (Arabic to English) word "Amid" to Persian, one could choose the character "ع" for "A" to specify this is originally an Arabic name, rather than the most common Persian character "ا" which is usually chosen for the English letter "A".

3.5 Transliterate or Not

Deciding on whether or not to translate or transliterate a name (or part of it) is a challenge for machine translation systems. Place names and organizations are the most common cases where both translation and transliteration can be necessary. For example, when a named entity such as "Alborz Mountain" is detected in a text, the first word needs transliteration and the second should be translated. In other words, the whole named entity is not transliterated. Another example is multi-word names; part of them may be a word with meaning which should not be translated. Languages such as Arabic and Persian that do not specifically mark a name in the text (for example with capital letters as in English) are more susceptible to such mistakes. For example the Arabic person name عبدالحميد (with one possible transliteration of "Abd al-hamid") is composed of two main components: "عبد" and "الحميد". The first, عبد, is an Arabic word meaning "worshipper". However, if this word is used as a person name, the system should not translate it.

Some studies in the literature (Al-Onaizan and Knight, 2002a,b; Chen et al., 2006; Hermjakob et al., 2008) investigate this problem specifically. In particular, Hermjakob et al. (2008) emphasize the positive effect of having a transliteration component for overall machine translation efficacy.

4. FORMULATION

To enable consistent explanations of the systems throughout this survey we define a framework for transliteration models and the systems that follow these models. The formulations for the transliteration process, and transliteration model, apply primarily to generative transliteration methods; discussion of bilingual transliteration corpus and evaluation applies equally to generative and extractive transliteration approaches.

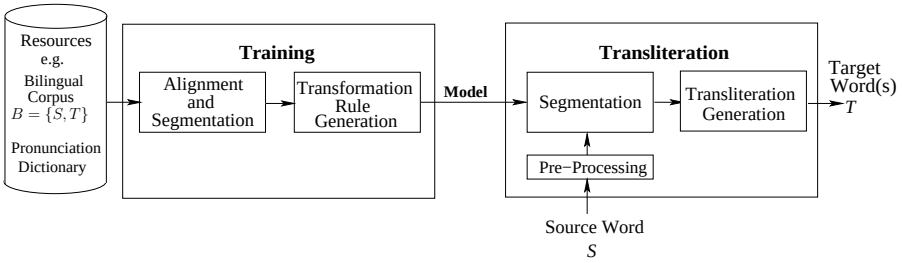


Fig. 2. A general framework for a generative transliteration system.

4.1 Transliteration Process

The generative transliteration process usually consists of two phases: a *training stage*, running on a bilingual training corpus $B = \{(S, T)\}$; and, a *transliteration generation* stage that produces one or more target words T for each source word S. This process is shown in Figure 2.

The training stage itself is composed of three tasks: alignment of source-target words at the character or sound level (explained below); segmentation (for example using graphemes or phonemes); and transformation rule generation. The transliteration stage consists of the segmentation of the new (test) source word, and target word generation. Not all existing approaches match this framework completely, but it is a close approximation for the majority of methods, including most recently proposed ones. We explain the differences between particular methods by comparing them with this general framework.

4.2 Alignment

In the training stage of a transliteration system, as shown in Figure 2, alignment of source and target substrings plays an important role in the accuracy that can be achieved by a model built on a given corpus. Figure 1, using vertical lines, shows the correspondence between the source and target substrings of some example transliterations. Such correspondence is discovered through alignment.

The quality of alignment can have a substantial effect on the overall performance of a transliteration system. In general, alignment can be considered at different representation levels. Broadly speaking these are:

—*Grapheme-based:* This category of methods finds the correspondence between the graphmes of each pair of source and target words. The alignment is based on minimizing the distance between the graphemes, for example using an Expectation-Maximization approach.

—*Phoneme-based:* Alignment between the graphemes of two words in a transliteration pair is carried out by making use of a common phonetic representation as a pivot. Since the available sounds in source and target languages may differ, the source and target words are not required to map to identical phonetic representations.

This high-level categorization has close parallels with concepts of different transliteration approaches, which we review in Section 5.

Different systems make use of different approaches for alignment: historically, approaches borrowed from machine translation have been popular. In translation, the order of tokens may change, so the alignment models are non-monotonous (that is, not order preserving). On the other hand, the order of graphemes or phonemes for transliteration is unlikely to change between two language representations. Monotonous alignment approaches have therefore been developed specifically for transliteration. Due to its historical importance and wide usage, we first provide a detailed explanation of machine-translation based alignment, and then survey transliteration-specific alignment approaches.

4.2.1 *Machine Translation-Based (Non-Monotonous) Alignmet.* Historically, machine transliteration has borrowed its alignment step from statistical machine translation (SMT)[5]. Statistical MT, in general, follows a sequence of modelling, parameter estimation, and search. An SMT model can be considered as a function of faithfulness to the source language, and fluency in the target language (Jurafsky and Martin, 2008). With the intention of maximizing both of these parameters, the fundamental model of SMT is generally defined based on a *translation model* (faithfulness) and a *language model* (fluency) as

$$P(S,T) = argmax_T P(S|T)P(T), \tag{1}$$

where S is a sentence in the source language, T is a sentence in the target language, $P(S|T)$ represents translation model, and $P(T)$ denotes target language model. The translation model represents the probability of T being a translation of S. The translation model itself consists of two parts: translation probability, and distortion probability. Distortion relates to re-ordering of the words in the translation process. For example, it is possible that a source word which was located at the beginning of a sentence, to be re-located to the end of a target sentence after translation. Although such re-ordering is quite common in translation, in transliteration such a phenomenon does not exist; by its nature transliteration preserves the sequence of the sounds of the source word in the target. The language model, $P(T)$, indicates the probability of having a string in the target language with the word-order generated in T.

The SMT model (Equation 1) is originally formed using Bayesian noisy channel theory (Brown et al., 1990). The assumption is that sentences in the target language are in fact in the source language but corrupted by noise. Therefore, we need a decoder that, given the sentence S, produces the most probable sentence T. The target model $P(T)$ specifies sentences that are valid in the target language, and the channel model $P(S|T)$ explains the influence of noise that changed the source sentence from S to T (Brown et al., 1990; Knight, 1999). As we explain in Section 5, Equation 1 is widely used in machine transliteration systems at the character or substring level (rather than word or phrase level). Therefore, when referring to this formula from Section 5 onwards, it will be in its use in the context of transliteration.

An important component of any SMT system is word alignment. Word-alignment is a mapping between the words of a pair of sentences that are a translation of each

[5]A comprehensive tutorial on alignment methods in statistical machine translation can be found in Knight (1999)

other. The most popular alignment methods are IBM Model 1, Model 3 (Brown et al., 1993), and the Hidden Markov Model (HMM) (Vogel et al., 1996; Toutanova et al., 2002; Och and Ney, 2003). IBM alignment models assume that word alignment is restricted to two sentences and does not propagate across sentence boundaries. In translation, there are some words in the source sentence that have no equivalent in the target, and there are some target words that are not specifically generated from any of the source words. Such phenomena can be modelled by considering NULL words in the source sentence, which are counted as sources of *spurious words* in the target sentence (Knight, 1999). Also, depending on the algorithm, alignments can be one-to-one, one-to-many, or many-to-many. Similarly in transliteration, as can be seen in Figure 1, a source character or a substring can be mapped to NULL (shown by not having a vertical line from source to target), one, or many characters in the target. Spurious characters are not common in transliteration. However, as explained in Section 3.3 on transliteration variants, it is possible to generate multiple target variants for a word where some transliterators may add extra vowels to make those variants easier to pronounce.

The parameters of IBM Models for any pair of languages are estimated using an EM (estimation-maximization) algorithm (Dempster et al., 1977). The EM algorithm finds maximum likelihood estimates of parameters in a probabilistic model, where the model depends on unobserved latent variables. EM alternates between performing an estimation (E) step, which estimates the likelihood by including the latent variables given the observed data and current estimate of the model parameters, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found in the E step. The parameters found in the M step are then used to begin another E step, and the process is repeated until parameter estimates converge (Dempster et al., 1977). The EM algorithm has been widely used in machine transliteration systems, mostly in the alignment step.

Many studies on machine translation use GIZA++ as their underlying word-by-word alignment system. Machine transliteration systems have also benefited from such alignment, performing it at the character level (AbdulJaleel and Larkey, 2003; Virga and Khudanpur, 2003b; Gao et al., 2004b).

GIZA++ is a SMT toolkit freely available for research purposes. The original program called *GIZA* was part of the SMT toolkit EGYPT, developed at the center of language and speech processing at Johns Hopkins University by Al-Onaizan et al. (1999) during a summer workshop. The extended version of this toolkit is called GIZA++ and was developed by Och and Ney (2003). It extends IBM Models 3, 4 and 5, alignment models using word classes, and includes: a HMM alignment model; more smoothing techniques for fertility, distortion, or alignment parameters; more efficient training of the fertility models; and more efficient pegging. Some previous work on transliteration employs a word alignment tool (usually GIZA++), as their word pair aligner (AbdulJaleel and Larkey, 2003; Virga and Khudanpur, 2003b; Karimi et al., 2006). Such studies have based their work on the assumption that the provided alignments are reliable.

4.2.2 *Transliteration Specific (Monotonous) Alignment.* Some transliteration systems (for example, Gao et al. (2004a,b); Li et al. (2004); Karimi et al. (2007)) focus

on alignment for transliteration — monotonous alignment versus non-monotonous alignment suitable for translation — and argue that precise alignment can improve transliteration effectiveness.

In older literature, alignment has been investigated for transliteration by adopting Covington's algorithm on cognate identification (Covington, 1996); this is a character alignment algorithm based on matching or skipping of characters, with a manually assigned cost of association. Covington considers consonant to consonant and vowel to vowel correspondence more valid than consonant to vowel. Kang and Choi (2000) revise this method for transliteration, where a skip is defined as inserting a null in the target string when two characters do not match based on their phonetic similarities or their consonant and vowel nature. Oh and Choi (2002) further revise this method by introducing *binding*, in which many-to-many correspondences are allowed. However, all of these approaches rely on manually assigned penalties that need to be defined for each possible matching.

More recently, phonological alignment has been proposed (Pervouchine et al., 2009). Here, linguistic knowledge of the phonetic similarity between two words is applied to measure the distance between phonemes directly, for example with reference to specified phoneme sets.

4.3 Transliteration Model

Transliteration transforms words from a source language to a target language. In general, such transformations are performed character by character, or substring by substring (where words are segmented using grapheme or phoneme boundaries).

Definition 4.1. A transformation rule is denoted as $\hat{S} \rightarrow (\hat{T}, p)$, where $\hat{S}$ is a substring of the source alphabet, $\hat{T}$ is a substring of the target alphabet, and p is the probability of transliterating $\hat{S}$ to $\hat{T}$. For any $\hat{S}$ appears as the head of n rules, $\hat{S} \rightarrow (\hat{T}_k, p_k) : \sum_{k=1}^{n} p_k = 1$.

We define a *transliteration model* as a method of forming transformation rules from training data. That is, patterns for segmenting source and target words, together with possible incorporated context knowledge applied on specific training data, define a transliteration model. Such models form the core of a *transliteration system* that, given a source word as input, generates a ranked list of possible transliterations as output. More formally, we define a transliteration system as follows.

Definition 4.2. A transliteration system M takes a source word S and outputs a ranked list L, with (T_j, pr_j) tuples as its elements. In each tuple, T_j is the j^{th} transliteration of the source word S generated with the j^{th} highest probability pr_j.

4.4 Bilingual Transliteration Corpus

Training and evaluation of transliteration systems require a bilingual corpus of source words and their transliterations (Definition 4.3). The number of acceptable transliterations of each source word can be greater than one; therefore, in the corpus specifications we define T as a set of *transliteration variants* available in the corpus for a given source word S.

Definition 4.3. A bilingual corpus B is the set $\{(S,T)\}$ of transliteration pairs, where $S = s_1..s_\ell$, $T = \{T_k\}$, and $T_k = t_1..t_m$; s_i is a letter, logogram, or symbol[6] in the source language alphabet, and t_j is a letter logogram, or symbol in the target language alphabet.

Such a corpus, however, is usually not readily available for transliteration studies, particularly for languages with few computerized resources.

4.5 Evaluation Metrics

Typical evaluation measures for machine transliteration are *word accuracy* and *character accuracy*. However, other metrics — such as mean reciprocal rank — are also used casually in the literature. Below, we divide evaluation schemes reported in the literature into two groups: single-variant and multi-variant metrics, based on their explicit consideration of transliteration variants in their formula.

4.5.1 *Single-Variant Metrics.* One of the standard transliteration evaluation measures is word accuracy. Word accuracy (A), also known as *transliteration accuracy* or *precision*, measures the proportion of transliterations that are correct:

$$A = \frac{\text{number of correct transliterations}}{\text{total number of test words}}.$$

In most studies, word accuracy is reported for different cut-off values. For example, TOP-1 word accuracy indicates the proportion of words in the test set for which the correct transliteration was the first candidate answer returned by the transliteration system, while TOP-5 indicates the proportion of words for which the correct transliteration was returned within the first five candidate answers. Consider a system that generates $L_{\text{system}} = \{A'XC', A'B'C'\}$ (ordered from left to right) for the source word $S = ABC$, for which accepted transliteration in the test corpus is $L = \{A'B'C'\}$. If TOP-1 word accuracy is calculated, then for the given source word the system does not increase its number of correct transliterations. However, if TOP-5 word accuracy is considered, then given that the second suggestion is correct, the system is awarded one point towards correct transliterations generated.

In general, the appropriate cut-off value depends on the scenario in which the transliteration system is to be used. For example, in a machine translation application where only one target word can be inserted in the text to represent a source word, it is important that the word at the top of the system-generated list of target words (by definition the most probable) is one of the words generated by a human in the test corpus. Alternately, in a cross-lingual information retrieval application, all variants of a source word in the target language might be required. For example, if a user searches for an English term "Tom" in Persian documents, the search engine should try to locate documents that contain both "تام" (3 letters: ت-ا-م) /tɒm/ and "تم" (2 letters: ت-م) /tɔm/, two possible transliterations of "Tom" that would be generated by human transliterators. In this case, a metric that counts the number of transliteration variants (T_k) that appear in the TOP-N elements of the system generated list, L, might be appropriate.

[6]Based on the script of the language (See Section 2, Writing Systems, for more information).

The second standard evaluation measure, character accuracy, is based on the edit distance between the system-transliterated word and the expected transliteration. The edit distance measures the number of character insertions, deletions and substitutions that are required to transform one word into another (Levenshtein, 1965). Character accuracy, or character agreement, checks for the percentage of matched characters for each word pair:

$$CA = \frac{len(T) - ED(T, L(T_i))}{len(T)},$$

where, $len(T)$ is the length of the expected target word T, $L(T_i)$ is the suggested transliteration of the system at rank i, and ED is the edit distance between two strings (Hall and Dowling, 1980). Note that in this formula we assume T contains only one variant (it can be generalized to more). When CA is used to evaluate a system, an average over all the test pairs is reported.

Some studies, in particular in transliteration extraction, report the *F-Measure*. It is calculated as the harmonic mean of *precision* (P) and *recall* (R):

$$F = \frac{2P \times R}{P + R},$$

where precision, similar to word accuracy, is percentage of correct transliteration pairs extracted, and recall is the percentage of correct pairs extracted over the total number of existing transliteration pairs in the collection (extracted or not).

Mean reciprocal rank (MRR) is an evaluation metric used in information retrieval studies to evaluate the ranked list generated by a search engine. The mean reciprocal rank, for machine transliteration, is the mean of the reciprocal of the rank at which the correct transliteration was generated, averaged over all the test words (Kantor and Voorhees, 2000):

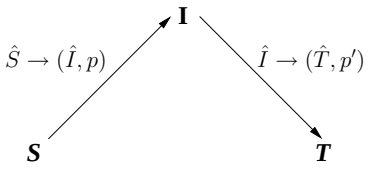
$$MRR = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \frac{1}{R_i},$$

where N is total number of test words, and R_i is the rank in L in which the ith test word has a correct transliteration.

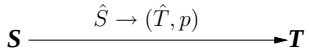
4.5.2 *Multi-Variant Metrics.* When more than one transliteration is available for a given source word when measuring effectiveness of the transliteration system, for example because of having a corpus created by multiple transliterators these multiple variants need to be taken into account. Hence, three varieties of word accuracy have been introduced: *uniform*, *majority* and, *weighted* (Karimi et al., 2007).

Uniform word accuracy (UWA) equally values all of the transliteration variants provided for a source word. Consider a word-pair (S, T), where $T = \{T_k\}$ and $|T| > 1$. Then a transliteration system that generates any of the T_k variants in T is successful.

For example, three English-Persian transliterators transliterate the name "Tom" and suggest the following transliterations {تام /tɒm/}, {تم /tɔm/}, and {تام /tɒm/}, respectively. Then, $T = \{$تام, تم$\}$ since there are two variants. For UWA both are equally valid as a correct transliteration.



(a) phonetic−based transliteration



(b) spelling−based transliteration

Fig. 3. A graphical representation of phonetic-based and spelling-based transliteration approaches, where I represents a phonetical representation of the source (S) and target (T) words.

Under majority word accuracy (MWA) evaluation only one of the provided transliterations is selected as valid. The criterion for choosing the preferred variant is that it must be suggested by the majority of human transliterators. In the example above, for the name "Tom", MWA considers $T = \{$تام /tɒm/$\}$ as the only valid transliteration.

Weighted word accuracy (WWA) allocates a weight to each of the transliterations based on the number of times that they have been suggested by multiple transliterators. In other words, all transliteration variants are valid but with a given weight. Continuing with the above example of "Tom", both variants are considered as valid, with transliteration تام /tɒm/ being two times more significant towards the accuracy score than تم /tɔm/.

Note the two MWA and WWA metrics differ from UWA only when duplicate transliterations are not removed from a testing corpus.

5. APPROACHES OF GENERATIVE TRANSLITERATION

Generative transliteration is the process of transliterating a given term (word or phrase) from a source language to a target language. That is, the script of the source term changes to the target language script, while the pronunciation is preserved as much as possible. Many different generative transliteration methods have been proposed in the literature, leading to much variation in terms of the proposed methodologies and languages supported. Due to the many varying attributes of these methods, such as the direction of transliteration, scripts of different languages, or different information sources, categorization of these studies is not straightforward.

In terms of direction of transliteration, *forward* and *backward* transliteration is introduced. Forward transliteration — or simply transliteration — is transliterating a word from one language to a foreign language. For example, forward transliteration of a Persian name "پروین" /pærviːn/ to English is "Parvin", and transliteration of the English place name "Melbourne" to Greek is "Μελβουρνη". Backward

transliteration or *back-transliteration* is transliterating an out-of-dictionary word from its transliterated version back to the language of origin. For example, back-transliteration of "Parvin" from English to Persian is "پروین", or "Μελβουρνη" from Greek to English is "Melbourne". Forward transliteration allows for creativity of the transliterator, whereas back-transliteration is strict and expects the same initial word to be generated.

Automatic transliteration has been studied between English and several other languages, including Arabic (Stalls and Knight, 1998; AbdulJaleel and Larkey, 2003; Sherif and Kondrak, 2007b; Freitag and Khadivi, 2007; Kashani et al., 2007), Persian (Karimi et al., 2006, 2007), Korean (Jeong et al., 1999; Jung et al., 2000; Kang and Kim, 2000; Oh and Choi, 2002, 2005), Chinese (Wan and Verspoor, 1998; Meng et al., 2001; Lin and Chen, 2002; Virga and Khudanpur, 2003b; Gao et al., 2004a; Zhang et al., 2004; Xu et al., 2006; Li et al., 2007; Jiang et al., 2007), Japanese (Knight and Graehl, 1998; Goto et al., 2004; Bilac and Tanaka, 2005; Oh and Choi, 2006a; Aramaki et al., 2007, 2008), and the Romantic languages (Lindén, 2005; Toivonen et al., 2005; Pirkola et al., 2006; Loponen et al., 2008). Transliteration approaches based on the script of languages can be classified into those methods proposed for languages with Latin script, languages with symbolic script, languages with Arabic script, and languages with Devanagari script for some of Indian languages. Most research for languages with similar scripts is devoted to cross-lingual spelling variants, and their application in search tasks. Transliteration between languages that are widely different in script is generally more challenging.

Another categorization of transliteration approaches is based on the information sources used in the process. This approach most clearly distinguishes the different methods proposed in the literature, and is the categorization that we follow. Based on information sources, transliteration systems can be categorized into:

—approaches that consider the task as a purely phonetical process and therefore use *phonetics*;

—approaches which perceive it as an orthographic process and use *spelling*;

—approaches that mix these two groups for a *hybrid* approach; and

—approaches that combine any number of the spelling- or phonetic-based methods (not both) for a *combined* approach.

We use the four categories of phonetic-based, spelling-based, hybrid, and combined to review the literature on generative machine transliteration. These different approaches and their performance are summarized in Tables I, II, and III at the end of this section. Note those tables show no direct comparison on effectiveness of the systems but rather show their variety in language scripts they deal with, their evaluation approach and corpora used, and their category of transliteration method.

5.1 Phonetic-based Methods

Most early studies on transliteration applied speech recognition methods, and studied transliteration in a phonetic-based framework. In the literature, this family of approaches is also called *pivot* or phoneme-based. The intuition behind this category of approaches is that phonetical representation is common among all

languages, which makes it possible to use it as an intermediate form between source and target languages (similar to interlingua MT). The other reason for the interest in phonetic-based approaches is the nature of the task; transliteration is a phonetical translation, and phonetics can capture the pronunciation of the words. A general diagram of a phonetic-based approach is shown in Figure 3 (a). Phonetic-based methods identify phonemes in the source word S, and then map the phonetical representation of those phonemes (I) to character representations in the target language to generate the target word(s) T. Different methods differ in their approaches of forming transformation rules ($\hat{S} \rightarrow \hat{I}$, and $\hat{I} \rightarrow \hat{T}$ based on Definition 4.1), and how phonemes or phonetical units of the source and target words are detected. We review these approaches in order of appearance, illustrating the main components of their generative transliteration system.

In general, phonetic-based systems borrow their transformation concepts from speech recognition phoneme-to-grapheme and grapheme-to-phoneme rule generation. Examples of such transformation rules for English spelling to phonetics, defined by Divay and Vitale (1997), are:

c → [k] / – {a,o},
c → [s].

This set of rules are read as: the grapheme "c" sounds as [k] if it is followed by "a" or "o", and it sounds [s] otherwise. Detecting phonemes of the words being processed is an important part of these systems, and directly affects the accuracy of these rules.

Arbabi et al. (1994) developed an Arabic-English transliteration system using knowledge-based systems and neural networks for pre-processing of the source Arabic words. Arabic names were input from an OCR (optical character recognition) system, which in turn was fed with phone-book entries. Given in Arabic script short vowels are generally not written, a knowledge-based system vowelized these names to add missing short vowels, and passed them to a neural network to determine whether they are reliable in terms of Arabic syllabification. If reliable, then these names were converted to their phonetical representation using fixed transformation rules stored in a table. The phonetical representation was then transformed to English script using another set of fixed rules. Comparing this system with the outline in Figure 2, there is no formal transliteration training component, and only one transliteration component exists that performs vowelization as a preprocessing of the source word S. The transliteration model was therefore predefined in the form of fixed transformation rules. The main drawback of this study is that the importance of forming transformation rules is ignored. The emphasis was vowelization of the names and separating Arabic and non-Arabic names through the syllabification process.

In contrast to the perception of Arbabi et al. (1994), the transformation rule generation task is non-trivial. Divay and Vitale (1997) investigated generation of phoneme-to-grapheme transformation rules (also known as sound-to-letter rules), its challenges, and applications. The pronunciation of words in any language is determined by many parameters. For example, the position of words (morphophonemics) can determine how they are pronounced; also elision or epenthesis can

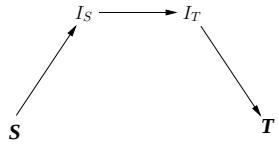


Fig. 4. A phonetic-based transliteration method. The intermediate step in Figure 3 (a) is expanded to cover phonetic representation of both of the source (S) and target (T) words.

make the pronunciation different from the orthographic presentation. Since the origin of proper names in languages can vary, the correspondence of the written names and their pronunciation can be very hard to specify, and in some cases they differ substantially from the spelling (Divay and Vitale, 1997). In some studies (Llitjos and Black, 2001; Huang et al., 2005; Huang, 2005), the problem of determining the diversity of proper names in terms of the ethnic groups that they belong to was studied by classifying them into their language group, or language family. This process increased the accuracy of systems that generate the grapheme-to-phoneme rules from proper names.

Knight and Graehl (1997, 1998) studied back-transliteration of Japanese out-of-dictionary words to English. Figure 3 (a) showed a general transliteration system that bridges between the languages using phonetical representation. In the phonetic-based approach proposed by Knight and Graehl (1998), four main steps were followed as shown in Figure 4. A Japanese source S was transformed to its phonetical presentation I_S, these source phonemes were then mapped to target English phonemes I_T, and a final phoneme-to-grapheme mapping generates the target English word T. Therefore, their model (derived from Equation 1, Section 4.2) was formulated as:

$$P(S,T) = \text{argmax}_T P(T) \sum_{I_S, I_T, S} P(S|I_S) P(I_S|I_T) P(I_T|T), \tag{2}$$

where $P(S|I_S)$ is the probability of pronouncing the source word, $P(I_S|I_T)$ converts the source sounds to the target sounds, $P(I_T|T)$ is probability of generating the written T from the pronunciation in I_T, and $P(T)$ is probability of sequence T occurring in the target language.

To perform the calculations, Knight and Graehl (1998) used a sequence of weighted finite-state transducers (WFSTs) and a weighted finite-state acceptor (WFSA). A finite state machine (FSM) is a model of behavior composed of a finite number of states, transitions between those states, and actions defined by performing the transitions. A weighted finite-state transducer is a kind of FSM which defines three parameters for each transition: input, output, and weight. A weighted finite-state acceptor has only one input symbol and a weight for each transition between the states, and specifies which output sequences are more probable than others. To match such a model with the transformation rules of a transliteration model (Definition 4.1), each transition can be considered as a transformation rule with the source and target mapping to input and output, with the associated probability mapping to a weight. In their model, Knight and Graehl (1998) implemented the language

model $P(T)$ using WFSA and the rest of the probabilities given in Equation 2 using WFST. To generate the best transliterations using WFSA, they implemented Dijkstra's shortest path and k-shortest paths algorithms (Eppstein, 1998) (TOP-K transliterations). The target language model implemented in $P(T)$ was a unigram word model made from the Wall Street Journal corpus, an on-line English name list, and an on-line gazetteer of place names. An English sounds inventory was taken from the CMU pronunciation dictionary. $P(I_S|I_T)$ was calculated based on frequency information taken from the alignment of 8,000 pairs of English and Japanese sound sequences learnt using the estimation-maximization (EM) algorithm (Dempster et al., 1977). In comparison to the base system in Figure 2, their system included both components, with WFSA and WFSTs built automatically and manually in the training stage, and then transferred as a transliteration model to the transliteration stage.

The English-Japanese model proposed in this study was strictly one-to-many so as to accommodate vowels that are often generated in a Japanese word after each occurrence of an English consonant (to avoid consonant clusters). In this model, mapping a sequence of characters in Japanese to only one English character is possible; this means that the model does not work in the reverse direction.

Knight and Graehl (1998) evaluated their automatic back-transliterator in two sets of experiments. One used 222 Katakana phrases; however, no evaluation result was reported for this experiment because they considered the task difficult to judge: some of the input phrases were onomatopoeic (words or terms that imitate the sound it is describing) and some were even difficult for humans to transliterate. The other experiment was on 100 names of U.S. politicians taken from Katakana. They compared their system's performance with four human transliterators — English native speakers — performing the same task. Human transliterators in general performed very poorly in comparison to the machine (24% versus 64% word accuracy). The reason for the low accuracy of humans, however, was their lack of knowledge of Japanese phonetics.

Stalls and Knight (1998) proposed a similar method for back-transliteration of Arabic out-of-dictionary words into English. The challenges for Arabic language is greater than for Japanese, as no specific pronunciation dictionary that covers out-of-dictionary words from different origins (not just English) is available, short vowels are not written in Arabic, and there is a general lack of electronic resources for Arabic pronunciation. The transliteration system was evaluated on a test corpus of 2,800 names that resulted in 32.1% accuracy. The study does not specify how many output words in the ranked list L (containing (T_j, pr_j) tuples, Definition 4.2) were considered in the evaluation. A reason for failure in back-transliterating some of the names was their non-English origin, which was not reflected in the pronunciation conversion models.

Wan and Verspoor (1998) investigated a method of English-Chinese transliteration using the general approach of transforming the English name to its phonetical representation, and transforming the phonetics to Chinese writing (Figure 3 (a)). Since the phoneme-to-grapheme process is considered the most problematic and least accurate step, they limited their model to place names only, to reduce variation. Since some place-names were partially translated, a pre-processing step (Fig-

ure 2) that performed dictionary look-up was used to detect those. A syllabification step segmented the English words to syllables, based on consonant boundaries. A sub-syllabification step further divided the syllables into sub-syllables to make them pronounceable within the Chinese phonemic set. Using a fixed English phoneme to Chinese mapping, the phonetic representation of each sub-syllable is transformed to Hanyu Pinyin, which is the most common standard Mandarin Romanization system. Pinyin uses the Latin script to represent sounds in standard Mandarin. Another fixed set of rules transforms Pinyin to Han (Chinese script). Therefore, the transliteration models were divided into a grapheme-to-phoneme step, and a phoneme-to-grapheme transformation which was based on a fixed set of rules. There was no evaluation reported for this approach.

Jeong et al. (1999) reported a method of back-transliteration for Korean out-of-dictionary phrases to English. Their study was divided into two main parts: identification of foreign words from Korean texts, and back-transliteration of them to English. The first step was extraction of non-Korean words using statistics of the phonetical differences between Korean words and transliterated words. In the second step, back-transliteration candidates were generated using a hidden Markov model (HMM) implemented as a feed-forward network with error-propagation. The transformation hierarchy was similar to Figure 3(a). That is, only one level of phonetical presentation was considered. The main formula for the ranking of the candidates was

$$\begin{aligned}
\mathtt{T} &= argmax_T P(T|S) \\
&= argmax_T P(t_1 t_2 \dots t_m | s_1 s_2 \dots s_l) \\
&= argmax_T P(t_1 t_2 \dots t_m) \times P(s_1 s_2 \dots s_l | t_1 t_2 \dots t_m) \\
&= argmax_T P(I_1 I_2 \dots I_m) \times P(s_1 s_2 \dots s_l | I_1 I_2 \dots I_m) \\
&\cong argmax_T \prod_j P(I_{t_j} | I_{t_{j-1}}) \times P(s_j | t_j),
\end{aligned}$$

where $\prod_j P(I_{t_j}|I_{t_{j-1}})$ shows the transition probability between two consecutive states in the HMM. In their model, Jeong et al. (1999) assumed that any Korean letter is only dependent on one single pronunciation unit in English. Their HMM model also considered only one-to-one relationships of characters. At the final stage, the candidate transliterations were compared against an English dictionary using similarity measures to prune the list of suggestions and rank them. They evaluated their transliteration accuracy in isolation and in an information retrieval framework. A bilingual corpus of 1,200 pairs was used by dividing that into training set of 1,100 pairs, and 100 for testing (no cross-validation). They reported TOP-1 accuracy of 47% and TOP-10 accuracy of 93%. The method resulted in 56% TOP-1 and 76% TOP-10 when dictionary matching was applied.

Jung et al. (2000) also proposed a method of English-Korean transliteration using an extended Markov window. They used the steps shown in Figure 3(a) where English word pronunciations were taken from the Oxford dictionary. A predefined set of rules then mapped the syllabified phonetic units to Korean. A heuristic method of syllabification and alignment was proposed to assign probabilities to the set of mapping rules (training stage). In the transliteration stage, they generated all

possible syllables of each English word based on the consonant and vowel boundaries in the equivalent phonetical shape (pre-processing and segmentation steps), then transliteration generation started. The transliteration model was based on an extended Markov window. Based on a general formula (derived from the joint probability of $P(S,T) = P(S)P(T|S)$) as:

$$\mathtt{T} = argmax_T P(S)P(T|S), \tag{3}$$

they incorporated the context in the target language into the probability calculations. Note this is different from the source-channel formula (Equation 1, Section 4.2) in that they use source language model rather than target. Using an extended Markov window, additional context around the source phonetic unit is included, and

$$\mathtt{T} = argmax_T \prod_i \frac{P(t_i|s_{i-1}t_{i-1})P(s_i|t_i s_{i-1})P(s_{i+1}|t_i s_i)}{P(s_{i+1}|s_i)}.$$

Their method was evaluated on a corpus of 8,368 English-Korean word-pairs with each English word accompanied with one or more transliterations. The results were reported for TOP-10 candidates generated, with a word accuracy of 54.9% when training and testing words were separated.

Oh and Choi (2002) studied English-Korean transliteration using pronunciation and contextual rules. In the training stage, English pronunciation units taken from a pronunciation dictionary were aligned to phonemes to find the probable correspondence between an English pronunciation unit and phoneme. Based on the pronunciation of the English word, a Korean word was generated. Word formation information in the form of prefix and postfix was used to separate English words of Greek origin. Their method is also referred to as *correspondence-based* transliteration (Oh and Choi, 2006a).

Lin and Chen (2002) presented a method of back-transliteration for English and Chinese. Their study however does not completely follow the generative transliteration framework in Figure 2 because of a learning process that modifies the transliteration model constantly. That is, a modified Widrow-Hoff learning algorithm automatically captures the phonetic similarities from a bilingual transliteration corpus. Their automatic method of extracting the phonetic similarities outperforms pre-defined phonetic similarities modelled as fixed transformation rules. In their approach, Lin and Chen (2002) used a pronunciation dictionary to transform both English and Chinese names to their IPA representation, and then applied a similarly measure on the phoneme (a similarity scoring matrix).

Virga and Khudanpur (2003a,b) examined English-Chinese transliteration using phoneme presentation of English names. They used the Festival speech synthesis system to convert English names into phonemes, extracted sub-syllables to match to Chinese pronunciations, and then converted these into Chinese. The approach they proposed was similar to Wan and Verspoor (1998), with a difference that the correspondence between English and Chinese pronunciations were automatically captured using GIZA++. They evaluated their method in retrieval of Mandarin

spoken documents from a topic detection and tracking (TDT) corpus using English text queries. There is no standard evaluation reported in their paper.

Gao et al. (2004a,b) investigated English-Chinese transliteration in a framework that did not follow the source-channel model, the most popular approach in the previous literature, and used a direct model (as opposed to indirect model as explained in the source-channel model in Equation 1). Comparing the two formulas of source-channel:

$$\mathtt{T} = argmax_T P(S|T)P(T), \tag{4}$$

and direct:

$$\mathtt{T} = argmax_T P(T|S), \tag{5}$$

they argue that the former concentrates more on well-formed target strings, but does not incorporate the neighboring phonemes, and also does not support many-to-one mapping between source and target phonemes (note this is also different from the model in Equation 3). They also investigated the target language model to the direct transliteration formula as

$$\mathtt{T} = argmax_T P(T|S)P(T), \tag{6}$$

to build their underlying model. They evaluated their model on 46,306 English-Chinese word-pairs extracted from LDC (Linguistic Data Consortium) named entity list using word accuracy and character accuracy metrics. Their results indicated that the direct model based on Equation 5 outperforms the source-channel model in their transliteration experiments using character accuracy. Using Equation 6 led to lower, but comparable results to the source-channel model.

5.1.1 *Discussion on Phonetic-based Methods.* In general, phoneme-based transliteration has a primary advantage of elevating the role of pronunciation in the transliteration process. However, the requirement for multiple steps in the process – including transformations from grapheme-to-phoneme, phoneme-to-grapheme, and sometimes phoneme-to-phoneme – increases the chance of propagating errors. Consider the general framework in Figure 2 and general steps of generative methods in Figure 3 and Figure 4 once more. The segmentation and alignment steps are never perfect with some errors introduced in particular in the segmentation steps. Increasing the number of steps that are sourced on this segmentation step also can introduce other errors to the system. For example, in the two steps of phonetic-based systems in Figure 4, one would expect $e_{S\rightarrow I} \times e_{I\rightarrow T}$ probable errors in total. For example, a generative transliteration system might generate hundreds of alternatives at each step, causing errors at an early stage to multiply and increase in significance during the later stages. Other sources of error could be the wrong choice of phoneme representation of a source or target substring. Note that these errors directly affect the final ranking of the suggested target words in general (the correct transliteration being ranked after an incorrect one), or can lead to totally wrong transliteration suggestion.

Another disadvantage of these methods is that they rely on bilingual pronunciation resources, which are not readily available for all languages.

5.2 Spelling-based Methods

While the main concern of phonetic approaches generally is finding the phonemes of the source word, substituting their phonetical representation, and then transferring them to the written target language, spelling-based methods map groups of characters in the source word S directly to groups of characters in the target word(s) T. These approaches are also called *direct* or grapheme-based in the literature. A general diagram of a grapheme-based approach is shown in Figure 3 (b). It can be seen that the number of steps in the transliteration process is reduced from two (or in some approaches, three) to one. Spelling-based approaches only rely on statistical information that is obtainable from the characters of the words. In this section, similar to phonetic-based approaches, we review the literature chronologically by order of appearance.

Kang and Kim (2000) proposed a method related to the phonetic-based approach of Jeong et al. (1999) for English-Korean transliteration and back-transliteration using a HMM. They approached the problem using the source-channel general formula in Equation 1 (Section 4.2). For their language model, they used the following bigram model:

$$P(T) = P(t_1) \prod_{i=2}^{m} P(t_i|t_{i-1}).$$

In this model $P(T)$ is approximated under a Markov first order dependence assumption. For each source word S all possible phoneme sequences are built; that is, the model does not rely only on one best segmentation, but generates all possible segmentations. Therefore, when there is no pronunciation available, those other segmentations are available and the source word has higher chance of being transliterated. Using all these segments, a network is created that can generate all possible transliterations of the source word. To assign a probability to each of these, substrings extracted from the training data were used. The size of each substring (called the phoneme chunk) was incorporated in the probabilities assigned to each transformation by multiplying by substring length. Evaluation was carried out using word accuracy and character accuracy metrics on an English-Korean corpus of 1,650 word pairs, with a fixed 150 test set separated; a second corpus of 7,185 word pairs; and a third corpus of 2,391 pairs. For English-Korean transliteration a maximum word accuracy of 55.3% TOP-1 was obtained while back-transliteration accuracy was 34.7%. Their second corpus resulted in a maximum of 58.3% word accuracy for forward transliteration, and 40.9% for back-transliteration. The third corpus was only used to evaluate the coverage of the system on the transliteration variants (explained in Section 3.3) in TOP-5 results.

Kang and Choi (2000) investigated English-Korean transliteration and back-transliteration using a new alignment algorithm and decision-tree learning. For English, 26 decision trees were learnt for each letter (26^2 decision trees), and for Korean 46 decision trees were learnt for each letter (46^2). Transformation rules in the decision trees used three past letters and three future letters as context for each

character in a source word. The system was evaluated on a bilingual transliteration corpus of 7,000 pairs, 6,000 being used for training and 1,000 used for testing. Word accuracy of 44.9% was obtained for transliteration using left and right context; for back-transliteration word accuracy was 34.2%. When information gain — a method of attribute selection for decision learning — was incorporated, the results improved accuracy to 48.7% and 34.7%, respectively.

AbdulJaleel and Larkey (2003) studied English-Arabic transliteration using n-gram models. Their transliteration was demonstrated with a transliteration model general formula (Equation 6), with the target language model being a bigram model: $P(T) = \prod_i (t_i|t_{i-1})$. Their system follows all the stages shown in the general generative system shown in Figure 2. Training aligns the word pairs from a bilingual transliteration corpus using GIZA++. Then, transformation rules are formed, and probabilities are assigned to these based on the frequencies in the corpus. The system was compared with a hand-crafted model that was constructed with carefully chosen rules as a baseline. Their system resulted in 69.3% TOP-1 word accuracy where the baseline hand-crafted system was 71.2% accurate, evaluated on a corpus of 815 word pairs taken from an Arabic corpus from AFP (Agence France Presse) newswire. The impact of transliteration was also evaluated in the context of a cross-lingual information retrieval task.

Zhang et al. (2004) and Li et al. (2004), in two studies which applied a similar approach, proposed a forward and backward transliteration method for the English-Chinese and English-Japanese language pairs. They investigated an orthographic alignment process to derive transliteration units from a bilingual dictionary. In their approach, alignment was introduced using the source-channel model as

$$P(S,T) = \prod_k P(< \hat{S}, \hat{T} >_k \mid < \hat{S}, \hat{T} >_{k-n+1}^{k-1}) \tag{7}$$

where $< \hat{S}, \hat{T} >$ represents alignment between two substrings of the source and target words. Each alignment k of $< \hat{S}, \hat{T} >_k$ in a sequence of alignments is approximated using its last n alignments $< \hat{S}, \hat{T} >_{k-n+1}^{k-1}$. On a corpus of 28,632 unique word pairs (transliteration pairs of different language origins, such as English, French, and Spanish), they reported various results in both studies when changing the context size, for different cut-off levels on the ranked transliterated outputs (TOP-1 and TOP-10). For example, a TOP-1 word error rate for English-Chinese transliteration when only unigrams were used was 46.9%. The word error rate is adverse of word accuracy (that is proportion of system errors to the total test words is calculated). Increasing the context had a positive impact on their results. The most prominent contribution of their work was integrating alignment into the main process of transliteration, which led to an optimized process compared to a system that separates these two steps.

Lindén (2005) investigated the problem of transliteration between Romantic languages, particularly for cross-lingual information retrieval. The approach used the model introduced in Equation 5 and considered past and future context in the source word to predict a target word character in an n-gram based framework:

$$P(T|S) = \prod_i P(t_i|s_{i-2}s_{i-1}s_i s_{i+1}).$$

This model was implemented using a weighted finite state transducer (WFST), and tested on 1,617 words in Finnish, Danish, Dutch, English, French, German, Italian, Portuguese, and Spanish. The system was evaluated using specially defined precision and recall metrics, making comparisons with other studies difficult. The Finnish data, however, was used only to check the robustness of the system and only added after the system was trained on other languages. The proposed model was particularly designed to extract cross-lingual spelling variants, and was therefore tested for such a task as well.

In a similar paradigm, Ekbal et al. (2006) investigated a revised joint source-channel based on an approach by Zhang et al. (2004) and Li et al. (2004) for Bengali-English. Transliteration units in the source word were chosen using a regular expression based on occurrences of consonants, vowels, and matra (a Bengali language writing delimiter). Differing past and future context (as in Equation 7), and context in the target word were examined. To account for one-to-many alignments between English and Bengali, hand-crafted transformation rules were provided to their system. In case of failure in alignment, even when incorporating handcrafted rules, manual intervention in the training phase was used to resolve the errors. Once the training was complete, the system was evaluated using a corpus of 6,000 people's names, with 1,200 for testing and 4,755 for training. Their best model achieved 69.3% TOP-1 word accuracy for Bengali-English, and 67.9% for back-transliteration.

Malik (2006) proposed a system of converting a word between two scripts of Punjabi: Shahmukhi, which is based on Arabic script, to Gurmukhi, which is a derivation of Landa, Shardha and Takri. The transliteration system used hand-crafted transliteration rules in two categories of character mappings and dependency rules. Dependency rules were contextual rules that define special cases of failure in simple character mappings. For evaluation, 45,420 words from classical and modern literature were extracted with an average transliteration accuracy of 98.95%.

Karimi et al. (2006) proposed *consonant-vowel based* algorithms for English-Persian transliteration. Their methods, named CV-MODEL1 and CV-MODEL2, were based on specific patterns of sequences of consonants and vowels in source words. Following the transliteration paradigm shown in Figure 2, in the training step, the word-pair (S,T) is first aligned (using GIZA++) to approximate the correspondence between source and target characters. A consonant-vowel sequence is then built for S by replacing each consonant with C and each sequence of vowels with V. This sequence together with the original characters are then broken into specific patterns such as CVC, CC, VC, and CV. Attaching the corresponding S characters to T characters based on these patterns, transformation rules are generated and the transliteration model is formed. The difference between the two methods, CV-MODEL1 and CV-MODEL2, is that CV-MODEL2 does not keep the original consonant characters in the final model, leaving them as C in the transformation rules that contain vowels. Note the consonants and vowels were orthographic rather than phonemic. Evaluation of these systems was conducted on an English-Persian transliteration corpus of 16,760 word-pairs, proper names from

different origins, created by 26 human transliterators. CV-MODEL1 and CV-MODEL2 resulted in word accuracy of 51.6% and 48.4% (TOP-1), respectively. A subset of English-only source words, 1,857 pairs, resulted in word accuracy of 61.7% and 60.0% (TOP-1), respectively, for CV-MODEL1 and CV-MODEL2.

Karimi et al. (2007) investigated both Persian-English and English-Persian transliteration by improving their *consonant-vowel* based methods (Karimi et al., 2006) in two ways: a new alignment algorithm that replaced GIZA++, and a new method of forming consonant-vowel sequences. The new alignment algorithm was based on *collapsed consonant-vowel* sequences of both source and target words, and the frequency of aligning their substrings. That is, sequences of consonants were aligned together, and similarly for vowels. These homogeneous sequences were broken into their constituent substrings based on the frequency of alignment in a training corpus. A similar concept, grouped consonants and vowels, was proposed for the transliteration stage where transformation rules, consistent with the alignment step, were formed using the boundaries of consonants and vowels. This method was named CV-MODEL3. Evaluations on a corpus of 16,670 English-Persian word-pairs from different origins showed a word accuracy of 55.3%, and for a sub-collection of all English source names it was 67.4%. Persian-English transliteration on a collection of 2,010 pairs led to 39.0% (TOP-1) word accuracy.

Sherif and Kondrak (2007b) investigated Arabic-English transliteration using dynamic programming and substring-based transducer approaches. To account for many-to-many mappings of source and target words that occur in transliteration (and had been ignored in the past studies), they applied phrase-based approaches of machine translation. Two methods were examined: monotone search using a Viterbi substring decoder, and a substring transducer. The advantages of the substring transducer are found to be its capability in implementing a word unigram language model, elimination of low probability mappings, and ability to handle mappings to NULLs implicitly and therefore reduces the confusion that NULLs may cause on the transducer. Their system was evaluated on a training corpus of 2,844 word pairs, 300 test word pairs; the language model was trained separately on 10,991 (4,494 unique) word pairs. Their results using word accuracy are reported only for seen data; that is, some of the training and testing data overlapped. Other studies also have used this evaluation paradigm. However, since the aim of a generative transliteration system is to transliterate unseen, newly appearing names, this method of evaluation seems unsatisfactory.

Li et al. (2007) proposed a transliteration method for personal names called semantic transliteration. The semantic aspect includes taking into account the language of origin, gender, and given or surname information of the source names. Their transliteration model was therefore formed as

$$P(T|S) = \sum P(T|S, l, g)P(l, g|S),$$

where l represents the language of origin and g represents gender. If any of the information was missing, then that source component was removed from their model. In their experiments three corpora were used with three languages of origin: Japanese, Chinese, and English. Names were separated to surname, female given name, and male given name. Using sequences of four characters, the origin of these

names and their gender were detected. Corpora used were reported with 30,000 pairs for Japanese-Chinese, 34,600 for Chinese-Chinese (Pinyin character dictionary for Chinese names), and 20,600 for English-Chinese. The performance of their system was reported using mean reciprocal rank, word accuracy, and character accuracy. The best overall accuracies achieved were 49.4% word accuracy, and 69.2% character accuracy. Although improvements were achieved in comparison to their baseline phonetic-based system, it is hard to compare the results with other studies which did not consider semantic information for English-Chinese transliteration. The main reason is lack of explanation on the source of the corpora used. A rough comparison to other studies on similar language-pairs (Zhang et al. (2004) and Li et al. (2004)) does not indicate a large difference in effectiveness.

5.2.1 *Discussion on Spelling-based Methods.* Spelling-based approaches for transliteration aim to model a direct mapping from group of characters in a source word to characters in a target word. The most widely-used technique include source-channel and language-model approaches. In comparison to phonetic-based approaches, spelling-based techniques reduce the number of steps involved in transliteration and can thereby remove some potential sources of error in the overall process. These methods expect to be trained well on all different source-target transformation rules (substring level) to be able to provide any correct transliteration, whereas in phonetic-based methods a pronunciation dictionary should cover all these conversion rules to the system.

5.3 Hybrid Methods

The phonetic-based and spelling-based transliteration approaches reviewed in the previous sections were investigated as two separate categories. Researchers have also considered a combination of these two categories as a third option (Figure 5). Phonetic-based approaches, having extra steps, are in general reported to be more error-prone than their spelling-based counterparts, and typically the success rates of purely phonetic-based approaches are lower than spelling-based methods, particularly for Arabic script languages that lack written short vowels. However, although spelling-based methods are more successful than phonetic-based approaches overall, they are less able to handle words where pronunciation differs widely from the spelling. For example, the English place name "Edinburgh" is pronounced /'ɛdnbɜrə/ with "gh" sounds different from its normal pronunciation. In this section, an overview of hybrid approaches is reported; these aim to incorporate the strength of each category for higher overall accuracy.

Al-Onaizan and Knight (2002a,b) studied Arabic-English transliteration using both phonetic and spelling information. The hybridization is based on a linear combination of the probabilities of these two methods:

$$P(T|S) = \lambda P_s(T|S) + (1 - \lambda)P_p(T|S), \tag{8}$$

where $P_s(T|S)$ represents the probability given by spelling approach and $P_p(T|S)$ is the score from phonetic approach and λ is a tunable weight parameter ($0 \leq \lambda \leq 1$). Their system follows the scheme shown in Figure 5. The spelling approach (M_s) followed the source-channel model using Equation 4. The phonetic component (M_p) was adapted from Stalls and Knight (1998). The implementation of Equation 8

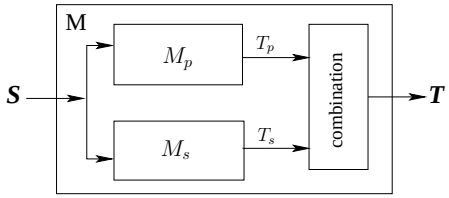


Fig. 5. A general hybrid transliteration approach that combines a phonetic-based (M_p) and a spelling-based (M_s) system. The combination method differs for different studies.

corresponds to the *combination* box in Figure 5. This approach was only proposed and evaluated for names of people, however. Names of locations and organizations, which can be partly translated and partly transliterated, were handled differently. Evaluations showed improvement (11.9% in comparison to a phonetic-based method, but a decline of 3.7% in accuracy in comparison to spelling-based method) in word accuracy using a hybrid method over phonetic-based methods in the first suggestion of the transliteration system (TOP-1).

Bilac and Tanaka (2004a,b, 2005) demonstrated that back-transliteration accuracy can be improved by direct combination of spelling and pronunciation information. The difference of their work from the method proposed by Al-Onaizan and Knight (2002a,b) is that, rather than producing back-transliterations based on spelling and phonetics independently, and then interpolating the results, they performed the combination during the transliteration process of each source word. In other words, the system M in Figure 5 is modified to generate target substrings one at a time, T_{kp} and T_{ks}, from the phonetic- and spelling-based generative components (M_p and M_s) instead of a whole word. They therefore proposed the following formula for the combination component of a hybrid method:

$$P(\hat{T}_k|\hat{S}_k) = \lambda P_s(\hat{T}_k|\hat{S}_k) + (1-\lambda)P_p(\hat{T}_k|\hat{S}_k).$$

Then, using the source channel formula in Equation 4, they scored the transliterations for a ranked output. In their system, the alignment was performed using the EM algorithm and following the Al-Onaizan and Knight (2002a,b) approach; the underlying transliteration model was kept as a WFST.

Evaluation of this system was performed on back-transliteration of Japanese and Chinese out-of-dictionary terms to English. A bilingual transliteration corpus taken from the EDICT Japanese-English dictionary (Breen, 1993), including 714 word pairs with known pronunciations, was used. The results showed 84.6% TOP-1 accuracy for this corpus (without language model). Another corpus was taken from Katakana comprising 150 word pairs with pronunciations extractable from the CMU dictionary; this resulted in 38.0% TOP-1 accuracy in comparison to 38.7% for the phonetic-based approach, and 32.7% for the spelling-based approach (without language model). Using a language model in their experiments resulted in small or no improvements. In general, evaluation on both Japanese and Chinese transliterations showed that direct combination for certain corpora can increase accuracy.

Oh and Choi (2005), and Oh et al. (2006b,c) investigated a method of hybridization of spelling- and phonetic-based approaches for English-Korean and English-Japanese transliteration. They criticized the hybrid models introduced so far for ignoring the dependence of the source word graphemes and phonemes whereas Oh and Choi (2002) had considered this relation in their correspondence-based method. Other criticisms of the previous hybrid models were that they assigned a fixed weight to each of the spelling or phonetics approaches whereas, depending on the source word, some are transliterated more phonetically and some are more based on the spelling. They therefore approached the transliteration problem by combining the spelling and phonetics, with consideration of correspondence information, in one model. Three machine learning algorithms were implemented to bring all these methods to one framework: a maximum entropy model, decision-tree learning, and memory-based learning. Transformation rules were learned using all the approaches (phonetics, spelling, correspondence, and a hybrid of phonetics and spelling) with a context length of three on each side of the transliteration unit that is mapped to the target substring.

Their evaluation results showed improvements in word accuracy in comparison to each of the other models independently. For English-Korean, word accuracy was 68.4% for a corpus of 7,172 word pairs where 1,000 were chosen for testing. English-Japanese transliteration resulted in 62.3% word accuracy.

5.3.1 *Discussion on Hybrid Methods.* Hybrid approaches for transliteration combine both spelling-based and phonetic-based information into one system to generate candidate target words. Different techniques include calculating the probabilities separately to each information source and then merging them together, or taking the interdependencies of spelling and phonetics of words into account directly in the transliteration model. For some test corpora, hybrid methods have shown significant improvements over single sources of evidence.

5.4 Combined Methods

System combination schemes have been shown to be successful for different natural language processing applications such as machine translation (Nomoto, 2004; Matusov et al., 2006; Rosti et al., 2007), part-of-speech tagging (Roth and Zelenko, 1998; van Halteren et al., 1998), speech recognition (Paczolay et al., 2006; Gales et al., 2007), parsers (Henderson and Brill, 1999; Nowson and Dale, 2007), word-sense disambiguation (Pedersen, 2000), text categorization (Larkey and Croft, 1996), and information extraction (Banko and Etzioni, 2008).

Combining multiple systems is usually performed in one of the two frameworks: *glass-box*, or *black-box* (Huang and Papineni, 2007). *Glass-box* combination occurs when systems use details of their internal functionality in the combined system; hence combination happens before any final output is generated. An example of such a method for machine transliteration would be the linear combination of spelling- and phonetic-based methods as explained in Section 5.3 (in particular, Bilac and Tanaka (2004a,b, 2005)), under hybrid methods literature. These approaches often showed improvements in the effectiveness of the transliteration systems in comparison to using spelling-based or phonetic-based approaches individually.

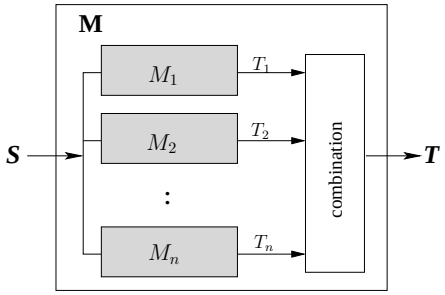


Fig. 6. A general combined (black-box) transliteration approach that combines multiple systems ($M_i; i = 1...n$) into one (M).

On the other hand, *black-box* combination works on the outputs of the systems, while the internal function of the systems is not altered (Huang and Papineni, 2007). This method has been repeatedly applied in machine translation. Generally, combining systems is advantageous for two reasons: first, systems errors are independent and they do not propagate to each other (Bangalore et al., 2001); and second, each of the systems has its own efficacy, and combining accumulates these to the final system. However, weak methods may dilute the performance of the final system, so system selection is crucial.

Combined approaches have only recently been introduced to machine transliteration, leaving room for further studies. Oh and Isahara (2007a) studied English-Korean and English-Japanese transliteration using a combination of transliteration systems. They proposed a method based on support vector machines (SVM) and maximum entropy models (MEM) to re-rank the outputs of individual transliteration systems. These individual systems were from a variety of spelling-based, phonetic-based, and hybrid methods. Both machine learning components, SVM and MEM, were trained using confidence score, language model, and Web frequency features. A confidence score was the probability (pr_j) assigned to each generated target word T_j in the list of (T_j, pr_j) candidate transliterations that each system produced. However, it is not clear that these scores are in fact comparable across different systems. The Web frequency parameter was adapted from other Web-based systems, similar to the method proposed by Al-Onaizan and Knight (2002b) which counts the co-occurrence frequencies of the transliteration pair on the Web.

For evaluation of their combined method, Oh and Isahara (2007a) used two corpora. An English-Korean corpus consisting of 7,172 pairs, and an English-Japanese corpus consisting of 10,417 pairs from the EDICT dictionary. Both corpora contained proper names, technical terms, and general terms. In their experiments, a fixed set of training and testing sub-corpora were used for evaluations (no cross-validation). Using seven individual systems, they reported 87.4% (TOP-1) word accuracy for English-Japanese transliteration, and 87.5% for English-Korean, when the MEM-based approach is used. For the SVM-based approach these results were 87.8% and 88.2%, respectively.

Karimi (2008) proposed a combined transliteration method in a black-box framework similar to the scheme shown in Figure 6. Multiple spelling-based translit-

eration systems ($M_i; i = 1...15$) were aggregated into one system M with the combination method being a mixture of a Naïve-Bayes classifier and a majority voting scheme. The system was evaluated for both English-Persian and Persian-English. English-Persian was trained and tested on a controlled corpus of 1,500 English words transliterated by seven human transliterators. Persian-English was evaluated on a corpus of 2,010 Persian person names accompanied by variants of their possible English transliterations. Experiments using ten-fold cross-validation of these corpora led to 85.5% word accuracy (UWA) for English-Persian and 69.5% word accuracy (UWA) for Persian-English, significantly improving the performance of the best individual system.

5.4.1 *Discussion on Combined Methods.* Combined methodologies to transliteration bring together a number of independent systems outputs and combine these into a single list of candidate answers. The key idea is that the unique strengths of each individual system can be combined to give a better overall answer. Such techniques have only recently began to be investigated for machine transliteration, but initial results are promising.

5.5 Transliteration Evaluation

From the studies reported in the previous sections it can clearly be seen that the performance of transliteration approaches are evaluated using bilingual transliteration corpora (Definition 4.3, Section 4.4). Traditionally, the transliteration pairs in such corpora are extracted from bilingual documents, dictionaries (Knight and Graehl, 1998; AbdulJaleel and Larkey, 2003; Bilac and Tanaka, 2005; Oh and Choi, 2006a; Zelenko and Aone, 2006), or gathered explicitly from human transliterators (Al-Onaizan and Knight, 2002a; Zelenko and Aone, 2006; Karimi et al., 2006, 2007). Some evaluations of transliteration methods depend on a single unique target word for each source word, while others take multiple transliterations for a single source word into consideration.[7]

The effects of corpus composition on the evaluation of transliteration systems had not been specifically studied till 2007 (around a decade after the first machine transliteration studies appeared), with only implicit experiments or claims made in the literature regarding the effects of introducing different transliteration models (AbdulJaleel and Larkey, 2003), or language families (Lindén, 2005), or application-based effectiveness (for example, for cross-lingual information retrieval (Pirkola et al., 2006)).

Karimi et al. (2007) were the first to investigate the effects of corpus construction on the reported effectiveness of transliteration systems. In their study the number of transliterations for each source word, the prior language knowledge of human transliterators used to construct the corpus, and the origin of the source words that make up the corpus were controlled. Their experiments — on a corpus of 1,500 English-Persian pairs from three different origins (English, Dutch, and Arabic; 500 each) transliterated by 7 different human transliterators — showed that the word accuracy of machine transliteration systems can vary by up to 30% depending on

[7]For many past studies it is difficult to determine which categories they fall in, due to a lack of explanation on the data that has been used.

the corpus on which they are run. The main reason is shown to be low agreement between human transliterators: 33%.

In addition to computing agreement, Karimi (2008) also investigated the transliterator's perception of difficulty of the transliteration task with the ensuing word accuracy of the systems. Interestingly, when using corpora built from transliterators that perceive the task to be easy, there is a large difference in the word accuracy between the two systems, but on corpora built from transliterators who perceive the task to be more difficult, the gap between the systems narrows. Hence, a corpus applied for evaluation of transliteration should either be made carefully with transliterators with a variety of backgrounds, or should be large enough and be gathered from various sources so as to simulate different expectations of its non-homogeneous users.

Overall, to prevent incorrect judgements over different systems, Karimi et al. (2007) and Karimi (2008) recommended four to five human transliterators with different backgrounds to be involved in corpus construction to keep the ranking and perceived accuracy of the systems stable over different corpora. Also, it was found that if only a single target word is available for every source word, then evaluation results for one corpus are unlikely to translate to other, except in rare cases where human transliterators are in 100% agreement for a given language pair.

Therefore, given the large variations in system accuracy that are demonstrated by the varying corpora (as shown in (Karimi et al., 2007)), we recommend that firstly, extreme care be taken when constructing corpora for evaluating transliteration systems, and secondly, studies must give complete details of their corpora.

5.6 Summary of Transliteration Generation

To summarize the literature on generative machine transliteration, phonetic-based approaches were popularized in 1990s when the first papers on automatic transliteration were published. These approaches evolved over the years, but their demand for pronunciation resources and language-dependant grapheme-to-phoneme or phoneme-to-grapheme conversion systems made them less appealing. Spelling-based approaches, on the other hand require fewer linguistic resources, have been more successful. Combining the two approaches has led to mixed results.

A general overview of the methods, corpora used, and accuracies reported in the reviewed literature is shown in Tables I, II, and III for handcrafted transformation rule-based, phonetic-based, spelling-based, hybrid, and combined systems, respectively. We list selected performances from these studies to provide a short summary on the studies. However, as explained in the previous section on transliteration evaluation, the effectiveness reported in most of these studies is not directly comparable, unless the same corpora were used to conduct the evaluations.

As can be clearly seen from the summary tables (Tables I, II, and III), two important problems affect most studies: first, the corpus specifications are usually overlooked, with the majority reporting only the size of each corpus (explained in Section 5.5); and second, while word accuracy was the most-reported measure, some studies used other measures, making comparisons across studies difficult. For example, although most studies use word accuracy and character accuracy as their evaluation measure, there are others which use less conventional metrics such as error rate, and mean reciprocal rank.

Method/Language Script	Corpus Specification	Performance(%),Metric
Handcrafted Rules		
Arabic-English (Arbabi et al., 1994)	phone-book entries, size unknown	unreported
Shahmukhi-Gurmukhi (Malik, 2006)	words from literature, 45,420	99, word accuracy
English-Persian Persian-English (Karimi, 2008)	1,500, proper names, 7 variants 2,010, person names	56.2, word accuracy (UWA) 16.6, word accuracy (UWA)
Phonetic-based		
Japanese-English (b) (Knight and Graehl, 1998)	people names, 100	64, word accuracy
Arabic-English (b) (Stalls and Knight, 1998)	names (type unknown), 2,800	32, word accuracy
English-Chinese (Wan and Verspoor, 1998)	unreported	unreported
Korean-English (b) (Jeong et al., 1999)	type or source unknown, 1,200	56, word accuracy
English-Korean (Jung et al., 2000)	type or source unknown, 8,368	55, word accuracy
English-Korean (Oh and Choi, 2002)	type or source unknown, 7,185	52, word accuracy 92, character accuracy
Chinese-English (b) (Lin and Chen, 2002)	names (type unknown), 1,574	83, mean reciprocal rank
English-Chinese (Virga and Khudanpur, 2003b)	names (type unknown), 2,233 training 1,541 testing	13, character error rate (CA complement)
English-Chinese (Gao et al., 2004b)	LDC corpus, 46,306	36, word accuracy 77, character accuracy

Table I. An overview of accuracy of different transliteration methods in the literature (handcrafted rule-based and phonetic-based). Note the accuracies reported in this table are the best reported amongst all the experiments in the corresponding papers. (b) indicates back-transliteration. The performance of the reported methods are not directly comparable due to different evaluation metrics and corpora.

6. APPROACHES OF TRANSLITERATION EXTRACTION

In this section, we review the major studies in the field of transliteration extraction. Transliteration extraction is the process of discovering transliteration pairs from different multilingual resources, such as parallel and comparable documents, and the Web. Typically, transliteration extraction techniques share some translation extraction methods such as co-occurrence statistics; in addition, transliteration

Method/Language Script	Corpus Specification	Performance(%),Metric
English-Korean (f,b) (Kang and Kim, 2000)	type or source unknown, 7,185	58, word accuracy 41, word accuracy (b)
English-Korean (f,b) (Kang and Choi, 2000)	type or source unknown, 7,000	48, word accuracy 35, word accuracy (b)
English-Arabic (AbdulJaleel and Larkey, 2003)	extracted from AFP corpus, 815	69, word accuracy
English-Chinese, (Zhang et al., 2004)	type or source unknown, 28,632	error rate reported (word and character level)
9 European languages (Lindén, 2005)	dictionary, 1,617	70, precision defined using reciprocal rank
Bengali-English (Ekbal et al., 2006)	people names, 6,000	68, word accuracy
Arabic-English (Sherif and Kondrak, 2007b)	type or source unknown, 3,144	2.01, avg. edit distance
English-Chinese (Li et al., 2007)	different language origins, people names were gender separated	58, mean reciprocal rank 49, word accuracy 69, character accuracy
English-Persian Persian-English (Karimi et al., 2007)	1,500, proper names, 7 variants 2,010, person names	74, word accuracy (UWA) 53, word accuracy (UWA)

Table II. An overview of accuracy of different transliteration methods in the literature (spelling-based). Note the accuracies reported in this table are the best reported amongst all the experiments in the corresponding papers. (f) indicates forward transliteration, and (b) indicates back-transliteration. The performance of the reported methods are not directly comparable due to different evaluation metrics and corpora.

extraction considers phonetic or graphematic similarities. Automatic extraction of transliteration pairs can potentially:

—enrich the existing transliteration corpora with new pairs;
—alleviate human labour in corpus construction for generative studies; and
—add transliteration variants, such as regional variants (Kuo et al., 2009), to the existing pairs in a transliteration lexicon from various sources (for example Web documents).

A variety of methodologies are investigated in the literature, including using word co-occurrences (Nagata et al., 2001; Tsuji et al., 2002; Huang and Vogel, 2002; Lam et al., 2004; Oh et al., 2006a), phonetic similarity (Lam et al., 2004; Kuo and Yang, 2004; Kuo et al., 2007; Tao et al., 2006; Sproat et al., 2006; Lee et al., 2006b; Oh and Isahara, 2006; Kuo et al., 2008), and different machine learning techniques. Machine learning approaches used in transliteration extraction are from a variety of existing methods including both supervised and unsupervised learning, and different learning algorithms such as bagging, boosting (Chen and Hsu, 2008), active

Method/Language Script	Corpus Specification	Performance(%),Metric
Hybrid (glass-box)		
Arabic-English (Al-Onaizan and Knight, 2002b)	names of locations and organizations, unknown size	73, word accuracy
Japanese-English (b)	EDICT dictionary,714	85, word accuracy
Chinese-English (b) (Bilac and Tanaka, 2004b, 2005)	Katakana words, 150	38, word accuracy
English-Korean	7,172	68, word accuracy
English-Japanese (Oh et al., 2006c)	EDICT,10,417	62, word accuracy
Combined (black-box)		
English-Korean	7,172	87, word accuracy
English-Japanese (Oh and Isahara, 2007a)	EDICT,10,417 names, technical terms, and general terms	88, word accuracy
English-Persian	1,500, proper names, 7 variants	86, word accuracy (UWA)
Persian-English (Karimi, 2008)	2,010, proper names	70, word accuracy (UWA)

Table III. An overview of accuracy of different transliteration methods in the literature (hybrid and combined). Note the accuracies reported in this table are the best reported amongst all the experiments in the corresponding papers. The performance of the reported methods are not directly comparable due to different evaluation metrics and corpora. (b) indicates back-transliteration.

learning (Goldwasser and Roth, 2008; Kuo et al., 2008), and adaptive learning (Li et al., 2008; Kuo et al., 2008). Given the challenging nature of the task, combining different sources of evidence in the process of selecting candidate transliteration pairs has been popular in the literature, as we discuss in more detail in Section 6.2. A key example is the use of multilingual Web pages (Al-Onaizan and Knight, 2002b; Keskustalo et al., 2003; Oh et al., 2006a; Kuo et al., 2007; Wu and Chang, 2007).

A general, deliberately simplified, transliteration extraction system is shown in Figure 7. The functionality of such systems relies heavily on the available resources for a language pair, such as multilingual corpora, existing transliteration lexicons, and pronunciation dictionaries. Specifications of these resources are demonstrated below in Section 6.1. The extraction box in Figure 7 is different for each study based on their chosen methodology, described in Section 6.2.

We review the literature in two categories: first, some historic studies on major translation extraction that inspired later studies in transliteration extraction; and second, studies focused on transliteration extraction.

6.1 Transliteration Extraction Resources

A corpus may contain texts in a single language (*monolingual*), two languages (*bilingual* or *bitexts*), or multiple languages (*multilingual*). Bilingual and multilingual corpora can be parallel or comparable (non-parallel). There is some disagreement in the literature regarding the definitions of these two types of corpora (Pearson,

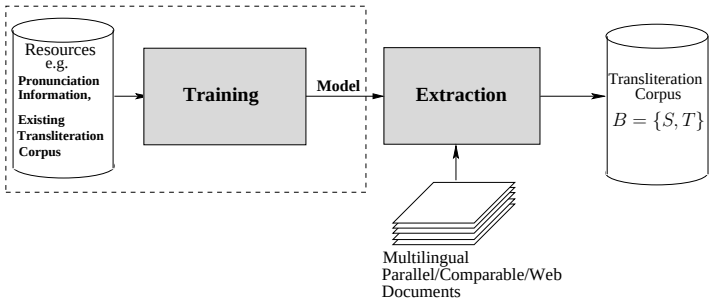


Fig. 7. A generic transliteration extraction system. The training on external resources, such as a pronunciation dictionary, or an already existing transliteration corpus (as seed), is shown as an optional component used by some previous work.

1998). In this survey, we use the following definitions that have been widely used in the recent literature.

Definition 6.1. A *parallel corpus*, in an ideal case, is a collection of texts in two or more languages. Each of the texts is an exact translation of one or more other languages, and the direction of the translation may be unknown.

Parallel corpora are attractive to researchers because of the opportunity of aligning translated texts (at the sentence, word, or even tag level (Smadja, 1992)), and because they can give insights into the nature of translation. However, the strict property of parallel corpora in providing exact translations makes them difficult to construct or obtain. In reality, however, most parallel corpora are not exact translations, mostly because of differences between languages, and difficulty in strictly following this condition when making such corpora.

Comparable corpora, as described in Definition 6.2, are another popular resource for computational lexicography and machine translation.

Definition 6.2. A *comparable corpus* is a collection of texts in two or more languages. The texts are similar, meaning that they contain similar information but they are not exact translation of each other. Generally, no information is available regarding the similarity.

Corpora, in general, provide the main knowledge-base for corpus linguistics: the study of language as expressed in its samples (in the form of a corpus), or real world text, to represent an approach to deriving a set of abstract rules by which a natural language is governed or relates to another language. The analysis and processing of various types of corpora is the subject of much work in computational linguistics, speech recognition and machine translation (McEnery and Wilson, 1996).

One of the recent challenges for corpus linguistics is using the World Wide Web as a corpus. Investigation of methods for culling data from the Web has introduced two approaches in corpus linguistics: "Web-as-corpus" and "Web-for-corpus-building" (Hundt, 2006). In the following section, we discuss this phenomenon for machine transliteration.

6.2 Literature on Transliteration Extraction

Learning *translation equivalents* from parallel or comparable corpora (Definitions 6.1 and 6.2) has been studied for machine translation for more than a decade. Statistical machine translation, in particular, is reliant on this process to learn translation by examples (Brown et al., 1990; Melamed, 2000). Proper names and technical terms in these texts need special attention; most of them rarely appear in documents and are therefore often undiscovered by methods that rely only on co-occurrence frequencies. As a result, transliteration extraction emerged as a study of methods of extracting transliteration terms, and consequently enriching translation lexicons.

Transliteration extraction studies in the 90s — formerly known and reported as named entity translation — were heavily influenced by machine translation techniques, especially statistical word alignment methods. Transliteration researchers, following the tradition of the MT community, started using parallel corpora. Later, the lack of parallel corpora — a rare resource for many languages — led to the exploration of approaches that benefit from comparable corpora, bilingual dictionaries, and nowadays, the Web. The main metrics of evaluation in the field of translation (and transliteration) extraction is *precision* (the percentage of correct correspondences that are found among from bilingual texts) and *recall* (the percentage of correct correspondences that are found over correct pairs existing in bilingual texts under process). Details of these studies are reviewed in this section; we first consider approaches for translation extraction for named entities, followed by transliteration extraction.

6.2.1 *Translation Extraction For Named Entities.* Brown et al. (1990) introduced sentence alignment on parallel corpora using a probabilistic transfer mechanism using the Expectation-Maximization (EM) algorithm. Other studies in machine translation (Brown et al., 1993) were subsequent. Gale and Church (1991) introduced word correspondence in parallel text. They argued that in aligned sentences, word order is not preserved and, therefore, the term *alignment* should only be used at the sentence level, and *correspondence* should be used at the word level. Replacing the probabilistic transfer dictionary that was used by Brown et al. (1990) with a contingency table, Gale and Church (1991) proposed using similarity measures (in particular ϕ^2, a χ^2-like measure) to find associations of words in aligned sentences. They applied their method on English-French data and used the morphological resemblance of these two languages to increase the precision of word correspondence.

Van der Eijk (1993) proposed the acquisition of bilingual lists of terminological expressions from a Dutch-English parallel corpus. The main strength of his work was considering phrases instead of words. Part-of-speech tagging, co-occurrence statistics, and the position of terms were used in this study. A similar approach was demonstrated by Kupiec (1993) for English and French using an annotated parallel corpus. He investigated an approach for finding multi-word correspondence by applying the Baum-Welch algorithm (Baum et al., 1970) on noun phrases found in aligned sentences. A heuristic method of disambiguation was used to resolve the problem of multiple senses for each noun-phrase in the source language.

Following the successful word and phrase alignment studies, translation of technical terms became a popular topic. The unfamiliarity of most translators with

domain-specific terms which often cannot be found in dictionaries motivated researchers to automatically extract those terms and their equivalents in other languages, and augment them to dictionaries. Dagan and Church (1994), in an early attempt at extracting transliteration equivalents from parallel corpora, developed a tool called *Termight* that semi-automatically extracts technical terms and their translations. This tool relied on part-of-speech (POS) tagging and word-alignment to extract candidate pairs, and the user was responsible for filtering. In this study, increasing recall, and therefore extracting less-frequent equivalents that word-aligners would miss, was the main goal.

In contrast to the previous studies on word alignment in parallel corpora, Rapp (1995) considered the correlation between the co-occurrences of words in non-parallel, comparable, English-German news documents. He showed that even in comparable texts, patterns of word co-occurrence strongly correlate. This study was the basis for further consideration of comparable corpora — instead of hard-to-find parallel resources — in the field, both for machine translation and transliteration.

Lexical co-occurrence information in comparable corpora was investigated in a study by Tanaka and Iwasaki (1996) for learning translation correspondence between words. Using a modified idea to that of Rapp (1995), they assumed that two words that co-occur in the source language preserve their co-occurrence in the target language. However, instead of a one-to-one correspondence, they introduced one-to-many relations between words. They implemented such an idea based on a stochastic translation matrix that was created using a steepest decent algorithm of linear programming.

Later, Fung and McKeown (1997) continued the trend of technical-term translation from noisy parallel documents (English-Japanese and English-Chinese): documents that had no potential of getting aligned at the sentence-level. Technical terms were extracted using the Justeson and Katz (1995) technical term finder based on the order of POS tags (technical terms are either adjective-noun or noun-noun multi-words). They proposed using dynamic recency vectors instead of absolute word positions in texts to find the similarity of the terms. Using these vectors, they formed spectral signals for all the words in texts, and then used pattern matching on those recognized signals as translations of the words. In their work, proper names and low-frequency terms were excluded. Although their work was focused on technical terms, no transliteration feature was considered.

Extraction of technical terms from the Web was also studied by Nagata et al. (2001). They proposed a method of technical term translation extraction for English and Japanese. Using partial translations of terms in Web documents — documents that contain translations of some phrases immediately after their first occurrence — they extracted a list of English-Japanese technical terms. The most important clue for distinguishing these partial translations were original words that occur in parentheses in front of their translations. Use of table aligned terms and term co-occurrence probabilities were also examined. Their experiments showed that mining the Web for technical term translations is particularly effective for the fields of computer science, aeronautics, and law.

Acquisition of transliterated proper nouns — not only technical terms — from a bilingual corpus was considered mainly using approaches based on phonetic

similarity, starting from late 90s and continuing until recently. The main criteria for differentiating these methods, after their learning algorithm, is their targeted language-pair, and the type of multilingual corpora used (parallel, noisy parallel, or comparable).

One of the first studies to target proper names in particular was by Collier and Hirakawa (1997) which focused on English-Japanese proper nouns from a noisy parallel corpus of news articles. They proposed a tool called NPT (nearest phonetic translation) which, using pre-defined rules, transforms Katakana words into all possible English string representations. substring matching implemented in their tool performs a search through the candidate English terms using dynamic programming algorithms, and finds the highest similar target term T to the source Katakana term S. Their algorithm was evaluated on a corpus of 150 aligned articles. They achieved 75% precision and 82% recall in their experiments.

Tsuji et al. (2002) attempted Katakana-French transliteration pair discovery by a method close to the approach explained by Collier and Hirakawa (1997). A set of transliteration rules — the transliteration model — was constructed using already existing pairs. Then using co-occurrence statistics in a noisy parallel corpus, candidate French words were found to form a transliteration pair. Evaluating their method on a corpus constructed from 21 news articles, they reported 80% precision and 20% recall.

Huang and Vogel (2002) investigated named entity translation of English and Chinese, with an emphasis on proper names (persons, locations, and organizations). In their proposed approach, they used a commercial named entity annotator system to extract named entities and their type from a parallel sentence aligned corpus. The candidate translations were chosen using co-occurrence frequencies and translation probabilities calculated based on the Brown et al. (1993) models. The technique was an iterative approach that started from initial annotations, refined annotations, and gradually created a dictionary of transliterations from the corpora. The main contribution of this work was its focus on named entities; however, it required a parallel corpus that is not easy to obtain for many languages.

6.2.2 *Transliteration Extraction Studies.* A completely different approach for transliteration discovery was pioneered by Al-Onaizan and Knight (2002b). Their method required neither parallel nor comparable corpora of bilingual documents. Although they still entitled their work with named entity translation, the transliteration nature of the task was considered and applied in the process. In order to build an Arabic-English named-entity dictionary, they first specified the named entity phrases and separated person names from location and organization names. A transliteration generation paradigm (that used both phonetic and spelling features) generated a ranked list of suggested transliterations. Candidates were then rescored using straight Web counts, co-references, and contextual Web counts. They also used Web search for unsuccessful transliteration generations. The evaluation however was small scale and used only 20 Arabic newspaper articles for testing and 21 for training.

Similar to the Al-Onaizan and Knight (2002b) study, in more recent literature much attention has focused on the Web as a resource of discovering both named-entity translation equivalents and transliteration-pairs (Masuyama and Nakagawa,

2005; Zhang et al., 2005; Chen and Chen, 2006). Different clues were considered to find transliteration equivalents on the Web; for example Lu et al. (2002) in a translation extraction study used anchor text linked to target language equivalents that lead to better named entity translation over purely statistical methods, and Oh and Isahara (2007b) validated the output of their generative transliteration systems using the Web. One other similar method was by Jiang et al. (2007). They investigated English-Chinese transliteration. Here the output of a phonetic-based transliteration system is combined with the transliterations mined from the Web for a given word to form a single ranked list of target words. Their generative step, a phonetic-based system, was based on the approach shown in Figure 4 (Section 5.1). The mining step queries the English source word in Chinese Web pages, and those words with maximum pronunciation similarity scores are extracted. The candidate Chinese words and the original English term are again queried to find the most frequently occurring pairs. One key advantage of using the Web is to provide rare transliterations that otherwise would not be captured by pure phonetic-based transliteration trained on a pronunciation dictionary. Their approach is different from similar methods (Al-Onaizan and Knight, 2002b) in its scoring formulas. They evaluated their results on a corpus from LDC with 25,718 pairs of which 200 were used for testing, 200 for development and the rest for training. They report their transliteration accuracy as 47.5% (TOP-1).

Other than Web-oriented approaches, traces of using phonetical and co-occurrence measures continue to be explored in recent literature. Lam et al. (2004), for example, used similarity of English and Chinese proper names at the phoneme level. Again, in case of failure, they resorted to the Web to extract transliterations. In later study (Lam et al., 2007) they argue that transliteration is both semantic and phonetic. That is, there is always the possibility of transliterating one part of a name semantically, and another part phonetically. They therefore proposed a named entity matching model that makes use of both semantic and phonetic evidence. The Web is also mined to discover new named entity translations from daily news, as an application of their matching algorithm. For the language pair of English-Chinese their approach shows effective discovery of unseen transliteration pairs.

Although phonetic matching was shown to be useful in the extraction task, a pronunciation dictionary may not include rare words, such as specific proper names. Lee and Chang (2003), therefore, investigated an extraction method similar to the previous studies but without the mapping of the source words to their phonetical representation. Using a parallel English-Chinese corpus, they first aligned sentences, then extracted the proper names in the source sentence. The candidate target words in the aligned sentence were then mapped to their corresponding source word using a recursive dynamic programming approach and some language-specific rules. Although this study claims to work on a general parallel corpus, the evaluations are only performed on corpora that do not conform to the definition of a parallel document corpus (Definition 6.1). Three corpora were used: a corpus of 2,430 transliteration word-pairs; a corpus of 150 test names versus a set of 1,557 potential transliterations; and a corpus of 500 English-Chinese dictionary entries. Evaluation metrics were average rank, average reciprocal rank (MRR),

word accuracy, character accuracy, and character recall. The reported average word precision was 86.0%.

At this point, the distinction between two groups of dominating methods became clearer: those which use direct generative models, and those which use phonetic conversion. A new technology was also used in different studies, mainly for logographic languages. The two steps of *recognition* and *validation* were introduced, which by and large were analogous to *training* for capturing a model, and *extraction* in Figure 7. Generally, recognition nominates transliteration pairs based on a model (which could be dynamically improved), and validation uses external resources to accept or reject the pairs.

As we have seen, the popularity of phonetic similarity-based approaches for transliteration acquisition, as a core technique, is included in almost any study that uses well-known types of resources: parallel, and comparable corpora. In a similar framework, other resources were also considered, however. For example, the use of search engine query logs for extracting transliteration pairs was examined by Brill et al. (2001). Brill et al. (2001) explored harvesting Katakana-English transliteration pairs from a widely used search engine query log gathered over one month for both Japanese and English. Their similarity measure was based on a noisy channel error model and edit distance concepts. They trained the model using 10,000 manually selected Katakana-English pairs, and used this to extract more pairs from queries submitted in Katakana and English scripts. To control for possible noise, such as misspellings, they only considered those Katakana terms that appeared more frequently than a specified threshold. For their initial 60,000 Katakana strings found from the logs, all possible English correspondents were extracted. Randomly selecting 1,500 pairs for testing, 97.5% were found to be correct matches.

Kuo and Yang (2004) criticized the approach of Brill et al. (2001) for its dependence on the availability of a large manually constructed transliteration lexicon to initiate the extraction process. They themselves used a small set of such resources as seeds for the training stage (200 manually selected English-Chinese transliterated pairs). The criteria in choosing the initial seed pairs was to have equal number of syllables in source and target words, to make the alignment of the syllables straightforward and easy for training. Their general mapping process obeys the steps proposed by Knight and Graehl (1998) (Figure 4, Section 5.1), explained in the transliteration generation section, in which both source and target terms should be transferred to a common phonetic representation based on a syllabification algorithm. In the modelling stage, the seed pairs are used to extract transliteration pairs from the English-Chinese Web pages crawled and filtered based on the criteria of having English and Chinese words in one sentence. The system gradually adds them into the initial set. Once the transliteration model is generated, the approach extracts transliteration pairs from the snippets of English-Chinese Web pages returned from querying of proper names on the Web. This study focused mainly on text-to-phoneme and phoneme-to-phoneme mapping, and also syllabification of the words, which makes the approach inapplicable for languages that may not have sufficient resources available to facilitate such processing of the words at the phonetic level. A minor modification of this work (Kuo and Yang, 2005) places more

emphasis on the use of confusion matrices and their positive effect on capturing additional transliteration pairs.

Subsequently, this work was presented more comprehensively with systematic evaluations. Kuo et al. (2007) explored transliteration extraction from Chinese Web documents that contain English words and, following two main steps of validation and recognition, extracted the most probable transliteration pairs. To extract possible Chinese candidates from texts, a k-neighborhood method was applied by inspecting a neighboring area of each source English word. For phonetic similarity, a noisy channel model was used. After generating a list of candidates, a hypothesis test was used to check whether a pair is suitable for being added to the final lexicon.

Oh and Choi (2006b), in an approach similar to Al-Onaizan and Knight (2002b), investigated the application of a transliteration generation scheme for extracting transliterated pairs for a domain-specific dictionary. Their method consists of three main steps: detection of candidate source words from texts using a HMM model; generating machine transliterations of the candidate terms; and matching the automatically generated target terms with words in the target language texts. Their evaluation was performed on a bilingual domain-specific dictionary and a manually constructed transliteration corpus.

The hybridization of different systems, similar to generative transliteration, was also explored for transliteration extraction. In a study to enrich an English-Korean transliteration lexicon, Oh et al. (2006a) combined three systems: a phonetic conversion model, a phonetic similarity model, and a corpus-based similarity model. The phonetic conversion model was defined as the category of systems that use the intermediate phonetic equivalents of two terms in their mapping step, while the phonetic similarity model referred to direct phoneme-based comparison of the words. The last model, corpus-based, used the co-occurrence frequency of a pair in a document corpus and on the Web. The hybrid method calculates the final similarity of a pair by taking the cube root of the product of the similarities by these three systems. To evaluate their system, Oh et al. (2006a) used a parallel corpus created from a bilingual Korean-English technical dictionary. To train each of their two phonetic-based systems they used an already available transliteration lexicon of 7,000 pairs. In their best setup, precision, recall, and F-score were 53.7%, 82.4%, and 65.2%, respectively. A maximum effectiveness improvement of 23% over individual systems was demonstrated.

Tao et al. (2006) investigated extraction of transliteration pairs from comparable corpora in an unsupervised framework, where both phonetic and temporal similarity were combined. Two main methodologies were applied in their study: first, the pronunciation score between two terms were calculated using a language-universal cost matrix for all the languages involved. Also, information on errors in second-language pronunciation, based on the differences in pronunciations in each language-pair was incorporated. Such information covers for what we called *missing sounds* in Section 3.2. Second, given that the link between the source and target documents might have been missed in comparable corpora, it can be hard to associate the corresponding information. In comparable corpora extracted from news articles, documents could be linked through their publish date. Tao et al. (2006) therefore used this information to calculate the correlation score

between frequency vectors of the named-entities of documents using the Pearson coefficient (Rodgers and Nicewander, 1988). Their method was evaluated for three language pairs: English-Arabic, English-Chinese, and English-Hindi. Evaluations showed that the combination of phonetic and temporal similarities is superior to using either of them in a linear combination framework (similar to Equation 8 for hybrid methods, Section 5.3).

Sproat et al. (2006) reported on named entity transliteration on a Chinese-English comparable corpus. In their method of extracting transliteration pairs, phonetical transliteration scores, a page-rank like scoring approach, co-occurrence frequencies, and temporal distribution (similar to Tao et al. (2006)) of candidate pairs were all used. The dataset contained 234 Chinese documents and 322 English documents. In a similar approach, Klementiev and Roth (2006) used phonetic and temporal distribution to match English-Russian named entities. They proposed discriminative alignment of substrings of words to match the transliteration variants. Their experiments are reported on a corpus larger than its pioneers; it contained 2,327 Russian and 978 English news articles. Another example of recent studies that consider comparable corpora for transliteration extraction is the research of Alegria et al. (2006) on Basque-Spanish. Transliteration rules (or transformation rules) were manually constructed, and scores computed based on such rules. They also considered scores from Web counts.

Lee et al. (2006b) studied transliteration extraction in the framework of a generative model that used the noisy channel approach. Transliteration pairs were aligned using the EM algorithm and dynamic programming. Their contribution was in defining novel similarity score functions. Although they attempted to avoid using heavily language-dependent resources such as pronunciation dictionaries and manual phonetic similarity rules, their algorithm was evaluated on a parallel corpus which is even harder to obtain. In comparison to earlier studies their experimentation and evaluation was comprehensive. Three corpora were used for evaluation: a bilingual English-Chinese proper name list; a bilingual dictionary; and a corpus of 300 aligned sentences. Word precision (or word accuracy), character precision (or character accuracy), and character recall (the number of correctly transliterated characters over total number of correct characters) were reported as evaluation metrics. Overall, word and character precision and character recall of 93.8%, 97.8%, and 97.5% were achieved, respectively.

In a follow up study, Lee et al. (2006a) adapted their previous proper noun transliteration extraction system to a more general task suitable for named entity translation extraction. They incorporated phrase translation and acronym expansion into their system, along with multiple knowledge sources. Similar to their previous study, a bilingual dictionary, a parallel corpus of English-Chinese sentences, and a transliteration lexicon were applied. Their proposed statistical named entity alignment system was compared to IBM Model 4 (Brown et al., 1993), a well-known statistical machine translation alignment model, and showed significant improvement in the alignment of English-Chinese names entities.

Similar to the previous studies by Al-Onaizan and Knight (2002b) and Oh and Choi (2006b), Oh and Isahara (2006) proposed a transliteration extraction method based on phonetic similarity and Web search. The novelty of this work is that Web

snippets — short summaries of documents returned in response to queries — were used. In terms of validating the transliteration pairs using the Web, their work is comparable to the studies by Brill et al. (2001) and Kuo and Yang (2004). The hypothesis was that when querying in Korean or Japanese, it is very likely that an English description of some of the technical terms or named entities will be obtained. The novel step uses both forward validation and backward validation, calculated based on the chi-square test. In forward validation they calculated $\sqrt[|S|]{Pr(T|S)}$; while in the backward direction, they swapped source and target words. Their evaluation results showed that using a joint validation leads to more accurate extraction of English-Japanese and English-Korean transliteration pairs.

Talvensaari et al. (2007) proposed a method for languages that share one script (Swedish and Finnish). They explored the use of skip-grams (Keskustalo et al., 2003) — or fuzzy matching — to align the words which were not found in a general-purpose dictionary. They were able to extract corresponding transliterations in a comparable corpus which other frequency-based and time-based methods had failed to discover. A more detailed study on European languages by Pirkola et al. (2007) investigated two main approaches of transliteration generation and extraction called TRT (transformation rule based translation) and FITE (frequency-based identification of translation equivalents). A combination of these two methods (FITE-TRT) was most successful when external resources, the Web and a multilingual dictionary, were used to verify the transliteration pairs.

Sherif and Kondrak (2007a) proposed a bootstrapping approach that used a stochastic transducer for Arabic-English transliteration extraction. For document-aligned named entity extraction, POS tagging was applied to 1000 English documents, and the list of names extracted from English documents was then refined manually. The performance of fuzzy matching and bootstrapping methods for a transliteration extraction task was unreported. Their transducer essentially learns one-to-one relationships, and overall the approach lacks context sensitivity.

Kuo et al. (2009) raised the importance of adding regional transliteration variants to a transliteration lexicon, that is different transliterations that are used in different regions of the world for the same source word. This is supportive of the challenges discussed in transliteration variants challenge in Section 3.3, and system evaluation in Section 5.5. In particular, Kuo et al. (2009) studied the English-Chinese language pair with Chinese being spoken in different countries, making the task a real life problem to study. Their transliteration extraction system was composed of a phonetic similarity model that is cross-trained on seed transliteration pairs collected from Web pages of different regions (China and Taiwan in their experiments) and then used to extract unseen pairs. Their results were promising, with the system being able to learn the region separated variants of source words.

6.3 Summary of Transliteration Extraction

Transliteration extraction is rooted in studies of finding translation equivalents in machine translation. The importance of considering proper names was only highlighted in more recent studies. Discovery of transliteration equivalents started with work based on parallel corpora, and then moved on to consider noisy-parallel texts. Usage of comparable corpora was introduced later, followed by a focus on the Web for a period of 3-4 years. Most recently, studies again tend to focus on

Study/Language Script	Method/Resource(s)	Performance(%),Metric
Japanese and English Collier and Hirakawa (1997)	phonetic similarity/NPC	82, precision 75, recall
English and Chinese Fung and Yee (1998)	information retrieval similarity measures/CC	unreported
Japanese and English Brill et al. (2001)	noisy channel model/TL, query logs	reported as a graph of number of extracted pairs
English and Chinese Huang and Vogel (2002)	co-occurrence frequency/PC	82, F-score
Arabic and English Al-Onaizan and Knight (2002b)	transliteration generation methods, co-occurrence, and Web count/WWW, TL	65, translation accuracy (TOP-1)
English and Chinese Lu et al. (2002)	anchor text/WWW	74, avg. inclusion rate (TOP-1) when combined the system with a dictionary
English and Chinese Lee and Chang (2003)	phonetic similarity/PC	86, precision
English and Chinese Kuo and Yang (2004)	phoneme-to-phoneme mapping/TL, pronunciation resource, Web pages	reported as a graph of number of extracted pairs
English and Chinese Lam et al. (2004)	phonetic similarity/CC	96, avg. reciprocal rank
English and Chinese Sproat et al. (2006)	phonetic similarity and temporal information/WWW	95, mean reciprocal rank on selected test corpus
English and Chinese Lee et al. (2006b)	phonetic similarity/CC	94, word accuracy 98, character accuracy
English and Korean Oh and Choi (2006b)	phonetic similarity and transliteration generation/ domain-specific dictionary, TL, pronunciation resource	99, precision 73, recall precision and recall are calculated on different corpora
English and Korean Oh et al. (2006a)	a hybrid method of phonetics similarity and corpus frequency and Web frequency/PC, TL, WWW	65, F-score
Swedish and Finnish Talvensaari et al. (2007)	fuzzy matching, skip-grams/ CC	
English-Chinese (Jiang et al., 2007)	Using the Web for ranking LDC corpus, 25,718 pairs 200 testing,200 development	48, word accuracy
English and Chinese Lam et al. (2007)	semantic and phonetic evidence/WWW	89, mean reciprocal rank
Arabic and English Sherif and Kondrak (2007a)	fuzzy matching/CC	75, precision
English and Chinese Kuo et al. (2007)	phonetic similarity/WWW	74, F-score
English-Russian English-Hebrew Goldwasser and Roth (2008)	active learning	71, recall 52, recall
English and Chinese Kuo et al. (2008)	phonetic similarity, active learning/ WWW, query results	83, precision 66, recall 74, F-score
English and Chinese Kuo et al. (2009)	phonetic similarity, Web pages of different regions	57, F-score

Table IV. An overview of the specifications of different transliteration extraction methods in the literature. General multilingual resources used in these studies are parallel corpus (PC), noisy parallel corpus (NPC), comparable corpus (CC), WWW, or a combination of these. Additional resources are monolingual corpus (MC), pronunciation dictionary (PD), and a sample transliteration lexicon (TL). The performance of the reported methods are not directly comparable due to different evaluation metrics and corpora.

comparable corpora in addition to the Web. The approaches taken by different researchers are somewhat similar in each of these periods of the evolution of this research topic. Transliteration characteristics, such as the phonetic resemblance of transliteration equivalents, were largely ignored in past machine translation studies. Now, transliteration has found its place as a separate topic of study, focusing on spelling and phonetical properties of the transliterated pairs.

A general overview of the reviewed transliterated-pair acquisition studies is listed in Table IV. This summary demonstrates that most studies are focused on a limited set of language pairs, mostly English and Chinese. Languages for which multilingual corpora are less readily available are hardly studied in the transliteration extraction area. Also, as with generative transliteration studies, evaluation is often inconsistent, and some studies introduce uncommon metrics that are not used by other researchers. This makes it difficult to compare their effectiveness, particularly when no standard corpus has been widely introduced or used so far.

7. SUMMARY

Machine transliteration, the process of transforming proper nouns and technical terms from one language to another while preserving pronunciation, has been developing as a field since the late 1990s. The machine transliteration literature can be classified into two main categories: generative transliteration, and transliteration extraction.

Generative transliteration approaches, which aim to directly map components (such as symbols or phonemes) of a source word to a target word, can be further classified into phonetic-based, spelling-based, hybrid, and combined approaches. The commonality is that all studies attempt to generate transliterated target words from original source words. Phonetic-based approaches take advantage of the fact that source and target words sound similar, and therefore map from one to the other via an intermediate phonetic representation. Spelling-based approaches omit this middle step, and statistically model direct mappings from source to target symbols. In general, spelling-based approaches tend to provide better performance, because the transliteration process consists of fewer steps than for the phonetic approaches. Additionally, there is less reliance on the availability of external linguistic data such as pronunciation dictionaries. Hybrid approaches combine both phonetic- and spelling-based evidence, while combined approaches aim to take advantage of multiple transliteration systems and merge these together into a single candidate result list. These techniques aim to leverage the strengths of individual approaches, combining them to give a more robust final answer. While the research into such combined transliteration approaches is in its infancy, the results of initial studies look promising, and the techniques are expected to lead to further gains in system performance in the future.

Transliteration extraction, in comparison to generative transliteration, aims to discover already transliterated pairs of words from a variety of multilingual resources, such as parallel and comparable corpora, and the Web. This approach is particularly valuable in finding transliteration pairs that can be used for training generative systems, which then allow previously unseen terms to be processed. Extraction of transliterated pairs has recently been a popular topic which, if suc-

cessful, could alleviate the need for manual compilation of large transliteration corpora for training of generative systems, and also be used as a method of enriching transliteration lexicons.

Transliteration has evolved from a sub-field of machine translation into an important research domain in its own right. While significant advances have been made over the last decade, some key challenges remain unsolved. As can be clearly seen from the presented survey of the transliteration literature, the evaluation of such systems presents on-going issues. Different studies use different corpora – often even for the same language pairs – and frequently do not provide sufficient details about the number of transliterators, the origin of the words, or the number of candidate answer transliterations that are used. Details about the use of training and testing data are often omitted. Furthermore, while a large range of evaluation metrics exist, these are inconsistently used in the literature, with most researchers reporting only a non-standard subset of performance statistics. These inconsistencies make it difficult to determine which developments in the transliteration field are significant advances, and make it near impossible to fairly compare the performance of different transliteration approaches in the literature.

To overcome these problems, we believe that the transliteration community needs to invest in the development of standard testbeds consisting of carefully constructed corpora (paying special attention to factors such as the origin of the words), together with common sets of terms for training and testing purposes[8] Although many studies focus on specific language pairs, most approaches are flexible enough to allow them to be run on different data sets. The availability and use of common evaluation frameworks has led to significant improvements in terms of the understanding of system performance in other fields such as information retrieval, through forums such as the Text REtrieval Conference (TREC), the European-language-based Cross Language Evaluation Forum (CLEF), and the NII Test Collection for IR Systems (NTCIR) which focuses on Asian languages. The transliteration community also needs to agree on a minimal set of standard evaluation metrics, and encourage their consistent use across different studies.

Machine transliteration is usually not an end in and of itself, but is often required in the context of other NLP type tasks, such as cross-lingual information retrieval and question answering, or machine translation. As such, it is also important to evaluate and demonstrate the usefulness of the different transliteration approaches in the context of different natural language applications; this is only done in few studies at present.

In terms of future system challenges, key developments are likely to evolve through the greater personalization of transliteration systems. This is expected to include automatic adaptation to specific dialects and regionalization within a particular language (for example, variant target transliterations for the same source word may be correct in different Arabic countries; future systems could therefore be driven by incorporating user context such as current location, or place of origin).

[8]Such attempt has been started recently (2009) by a shared task in Named Entities Workshop at ACL-IJCNLP 2009 (`http://www.acl-ijcnlp-2009.org/workshops/NEWS2009/index.html`, accessed 28 July 2009).

True multi-language transliteration is also a challenge: while some current systems aim to be flexible in their modelling approaches, the best performance tends to be achieved when language-specific considerations are incorporated. Techniques such as source language identification are therefore likely to lead to improved performance for broader transliteration systems that can be applied for many languages. Techniques for combining the output of different individual transliteration systems, which have only recently begun to be studied, are also likely to offer significant advantages.

Acknowledgements

NICTA is funded by the Australian government as represented by Department of Broadband, Communication and Digital Economy, and the Australian Research Council through the ICT centre of Excellence programme.

This work is also supported by an ARC discovery grant (Turpin).

The authors would like to thank the reviewers for their invaluable feedback on the earlier drafts of this paper.

REFERENCES

AbdulJaleel, N. and Larkey, L. S. 2003. Statistical transliteration for English-Arabic cross language information retrieval. In *Conference on Information and Knowledge Management*. New Orleans, Louisiana, 139–146.

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J., Melamed, D., Och, F. J., Purdy, D., Smith, N., and Yarowsky, D. 1999. Statistical machine translation. Tech. rep., Johns Hopkins University.

Al-Onaizan, Y. and Knight, K. 2002a. Machine transliteration of names in Arabic text. In *Proceedings of the ACL workshop on Computational approaches to semitic languages*. Philadelphia, PA, 1–13.

Al-Onaizan, Y. and Knight, K. 2002b. Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, PA, 400–408.

Alegria, I., Ezeiza, N., and Fernandez, I. 2006. Named entities translation based on comparable corpora. In *Proceedings of the EACL Workshop on Multi-Word-Expressions in a Multilingual Context*. Trento, Italy.

Aramaki, E., Imai, T., Miyo, K., and Ohe, K. 2007. Support vector machine based orthographic disambiguation. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation*. Skovde, Sweden, 21–30.

Aramaki, E., Imai, T., Miyo, K., and Ohe, K. 2008. Orthographic disambiguation incorporating transliterated probability. In *Proceedings of the International Joint Conference on Natural Language Processing*. Hyderabad, India, 48–55.

Arbabi, M., Fischthal, S. M., Cheng, V. C., and Bart, E. 1994. Algorithms for Arabic name transliteration. *IBM Journal of research and Development 38,* 2, 183–194.

Bangalore, S., Bordel, G., and Riccardi, G. 2001. Computing consensus translation from multiple machine translation systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding*. Kyoto,Japan, 351–354.

Banko, M. and Etzioni, O. 2008. The tradeoffs between open and traditional relation extraction. In *Proceedings of 46th Annual Meeting of the Association for Computational Linguistics: Human language Technologies*. Columbus, Ohio, 28–36.

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics 41,* 1, 164–171.

Bilac, S. and Tanaka, H. 2004a. A hybrid back-transliteration system for Japanese. In *Proceedings of the 20th international conference on Computational Linguistics*. Geneva, Switzerland, 597–603.

Bilac, S. and Tanaka, H. 2004b. Improving back-transliteration by combining information sources. In *Proceedings of First International Joint Conference on Natural Language Processing*. Lecture Notes in Computer Science, vol. 3248. Springer, 216–223.

Bilac, S. and Tanaka, H. 2005. Direct combination of spelling and pronunciation information for robust back-transliteration. In *Conferences on Computational Linguistics and Intelligent Text Processing*. Mexico City, Mexico, 413–424.

Breen, J. W. 1993. A Japanese electronic dictionary project (part 1: The dictionary files). Tech. rep., Monash Uinversity, Australia.

Brill, E., Kacmarcik, G., and Brockett, C. 2001. Automatically harvesting Katakana-English term pairs from search engine query logs. In *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium*. Tokyo, Japan, 393–399.

Brown, P. F., Cocke, J., Pietra, S. D., Pietra, V. J. D., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. 1990. A statistical approach to machine translation. *Computational Linguistics 16,* 2, 79–85.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computional Linguistics 19,* 2, 263–311.

Chen, C. and Chen, H.-H. 2006. A high-accurate Chinese-English NE backward translation system combining both lexical information and web statistics. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL on Main Conference Poster Sessions*. Sydney, Australia, 81–88.

Chen, C.-H. and Hsu, C.-C. 2008. Boosted voting for confirming synonymous transliteration. In *Proceedings of the IEEE International Conference on Information and Automation*. Changsha, China, 1337–1342.

Chen, H.-H., Lin, W.-C., Yang, C., and Lin, W.-H. 2006. Translating–transliterating named entities for multilingual information access. *Journal of the American Society for Information Science and Technology 57,* 5, 645–659.

Collier, N. H. and Hirakawa, H. 1997. Acquisition of English-Japanese proper nouns from noisy-parallel newswire articles using Katakana matching.

In *Proceedings of the 3rd Natural Language Pacific Rim Symposium*. Phuket, Thailand, 309–314.

Covington, M. A. 1996. An algorithm to align words for historical comparison. *Computational Linguistics 22,* 4, 481–496.

Crystal, D. 2003. *A Dictionary Of Linguistics And Phonetics*. Wiley-Blackwell.

Crystal, D. 2006. *How Language Works: How Babies Babble, Words Change Meaning, and Languages Live or Die*. The Overlook Press, New York.

Dagan, I. and Church, K. 1994. Termight: identifying and translating technical terminology. In *Proceedings of the 4th Conference on Applied Natural Language Processing*. Stuttgart, Germany, 34–40.

Dale, R. 2007. Language technology. In *Slides of HCSNet Summer School course*. Sydney, Australia.

Dempster, A., Laird, N., and Rubin, D. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society 39,* 1, 1–38.

Divay, M. and Vitale, A. J. 1997. Algorithms for grapheme-phoneme translation for English and French: applications for database searches and speech synthesis. *Computational Linguistics 23,* 4, 495–523.

Ekbal, A., Naskar, S. K., and Bandyopadhyay, S. 2006. A modified joint source-channel model for transliteration. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL on Main Conference Poster Sessions*. Sydney, Australia, 191–198.

Eppstein, D. 1998. Finding the k shortest paths. *SIAM J. Computing 28,* 2, 652–673.

Freitag, D. and Khadivi, S. 2007. A sequence alignment model based on the averaged perceptron. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Prague, Czech Republic, 238–247.

Fung, P. and McKeown, K. 1997. A technical word- and term-translation aid using noisy parallel corpora across language groups. *Machine Translation 12,* 1-2, 53–87.

Fung, P. and Yee, L. Y. 1998. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the 17th International Conference on Computational Linguistics*. Montreal, Canada, 414–420.

Gale, W. and Church, K. 1991. Identifying word correspondance in parallel texts. In *Proceedings of the workshop on Speech and Natural Language*. Pacific Grove, California, 152–157.

Gales, M., Liu, X., Sinha, R., Woodland, P., Yu, K., Matsoukas, S., Ng, T., Nguyen, K., Nguyen, L., Gauvain, J.-L., Lamel, L., and Messaoudi, A. 2007. Speech recognition system combination for machine translation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Honolulu, HI, IV–1277–IV–1280.

Gao, W., Wong, K.-F., and Lam, W. 2004a. Improving transliteration with precise alignment of phoneme chunks and using contextual features.

In *Information Retrieval Technology, Asia Information Retrieval Symposium*. Lecture Notes in Computer Science, vol. 3411. Beijing, China, 106–117.

Gao, W., Wong, K.-F., and Lam, W. 2004b. Phoneme-based transliteration of foreign names for OOV problem. In *proceedings of the 1st International Joint Conference on Natural Language Processing*. Lecture Notes in Computer Science, vol. 3248. Springer, 110–119.

Goldwasser, D. and Roth, D. 2008. Active sample selection for named entity transliteration. In *Proceedings of the 46th Annual Meeting of the ACL on Main Conference Poster Sessions*. Columbus, OH, 53–56.

Goto, I., Kato, N., Ehara, T., and Tanaka, H. 2004. Back transliteration from Japanese to English using target English context. In *Proceedings of the 20th International Conference on Computational Linguistics*. Geneva, Switzerland, 827–833.

Hall, P. A. V. and Dowling, G. R. 1980. Approximate string matching. *ACM Computing Surveys 12,* 4, 381–402.

Henderson, J. C. and Brill, E. 1999. Exploiting diversity in natural language processing: combining parsers. In *Proceedings of the Fourth Conference on Empirical Methods in Natural Language Processing*. College Park, Maryland, 187–194.

Hermjakob, U., Knight, K., and III, H. D. 2008. Name translation in statistical machine translation - learning when to transliterate. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human language Technologies*. Columbus, OH, 389–397.

Huang, F. 2005. Cluster-specific named entity transliteration. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. Vancouver, Canada.

Huang, F. and Papineni, K. 2007. Hierarchical system combination for machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Prague, Czech Republic, 277–286.

Huang, F. and Vogel, S. 2002. Improved named entity translation and bilingual named entity extraction. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*. 253–258.

Huang, F., Vogel, S., and Waibel, A. 2005. Clustering and classifying person names by origin. In *Proceedings of National Conference on Artificial Intelligence*. Pittsburgh, Pennsylvania, 1056–1061.

Hundt, M. 2006. *Corpus Linguistics and the Web (Language and Computers 59)*. Editions Rodopi BV.

Jeong, K. S., Myaeng, S. H., Lee, J. S., and Choi, K. S. 1999. Automatic identification and back-transliteration of foreign words for information retrieval. *Information Processing and Management 35,* 4, 523–540.

Jiang, L., Zhou, M., Chien, L.-F., and Niu, C. 2007. Named entity translation with web mining and transliteration. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*. Hyderabad, India, 1629–1634.

Jung, S. Y., Hong, S. L., and Paek, E. 2000. An English to Korean transliteration model of extended Markov window. In *Proceedings of the 18th Conference on Computational linguistics*. Saarbrücken, Germany, 383–389.

Jurafsky, D. and Martin, J. 2008. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. Prentice Hall.

Justeson, J. and Katz, S. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering 1,* 1, 9–27.

Kang, B.-J. and Choi, K.-S. 2000. Automatic transliteration and back-transliteration by decision tree learning. In *Conference on Language Resources and Evaluation*. Athens, Greece, 1135–1411.

Kang, I.-H. and Kim, G. 2000. English-to-Korean transliteration using multiple unbounded overlapping phoneme chunks. In *Proceedings of the 18th Conference on Computational Linguistics*. Saarbrücken, Germany, 418–424.

Kantor, P. B. and Voorhees, E. M. 2000. The TREC-5 confusion track: Comparing retrieval methods for scanned text. *Information Retrieval 2,* 2-3, 165–176.

Karimi, S. 2008. Machine transliteration of proper names between English and Persian. Ph.D. thesis, RMIT University, Melbourne, Australia.

Karimi, S., Scholer, F., and Turpin, A. 2007. Collapsed consonant and vowel models: New approaches for English-Persian transliteration and back-transliteration. In *The 45th Annual Meeting of the Association for Computational Linguistics*. Prague, Czech Republic, 648–655.

Karimi, S., Turpin, A., and Scholer, F. 2006. English to Persian transliteration. In *String Processing and Information Retrieval*. Lecture Notes in Computer Science, vol. 4209. Glasgow, UK, 255–266.

Karimi, S., Turpin, A., and Scholer, F. 2007. Corpus effects on the evaluation of automated transliteration systems. In *The 45th Annual Meeting of the Association for Computational Linguistics*. Prague, Czech Republic, 640–647.

Kashani, M., Popowich, F., and Sarkar, A. 2007. Automatic transliteration of proper nouns from Arabic to English. In *In Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages*. Stanford,California, 81–87.

Keskustalo, H., Pirkola, A., Visala, K., Leppänen, E., and Järvelin, K. 2003. Non-adjacent digrams improve matching of cross-lingual spelling variants. In *String Processing and Information Retrieval*. Lecture Notes in Computer Science, vol. 2857. Manaus, Brazil, 252–265.

Klementiev, A. and Roth, D. 2006. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*. Sydney, Australia, 817–824.

Knight, K. 1999. A statistical MT tutorial workbook.

Knight, K. and Graehl, J. 1997. Machine transliteration. In *Proceedings of the 8th Conference on European Chapter of the Association for Computational Linguistics*. Madrid, Spain, 128–135.

Knight, K. and Graehl, J. 1998. Machine transliteration. *Computational Linguistics 24,* 4, 599–612.

Kuo, J.-S., Li, H., and Lin, C.-L. 2009. Harvesting regional transliteration variants with guided search. In *Proceedings of the 22nd International Conference Computer Processing of Oriental Languages. Language Technology for the Knowledge-based Economy*. 133–144.

Kuo, J.-S., Li, H., and Yang, Y.-K. 2007. A phonetic similarity model for automatic extraction of transliteration pairs. *ACM Transactions on Asian Language Information Processing 6,* 2, 6.

Kuo, J.-S., Li, H., and Yang, Y.-K. 2008. Active learning for constructing transliteration lexicons from the Web. *Journal of the American Society for Information Science and Technology 59,* 1, 126–135.

Kuo, J.-S. and Yang, Y.-K. 2004. Constructing transliteration lexicons from web corpora. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*. Barcelona, Spain, 3.

Kuo, J.-S. and Yang, Y.-K. 2005. Incorporating pronunciation variation into extraction of transliterated-term pairs from Web corpora. In *Proceedings of the International Conference on Chinese Computing*. Singapore, 131–138.

Kupiec, J. 1993. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. Columbus, Ohio, 17–22.

Lam, W., Chan, S.-K., and Huang, R. 2007. Named entity translation matching and learning: with application for mining unseen translations. *ACM Transactions on Information Systems 25,* 1, Article 2.

Lam, W., Huang, R., and Cheung, P.-S. 2004. Learning phonetic similarity for matching named entity translations and mining new translations. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Sheffield, UK, 289–296.

Larkey, L. S. and Croft, W. B. 1996. Combining classifiers in text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Zurich, Switzerland, 289–297.

Lee, C.-J. and Chang, J. S. 2003. Acquisition of English-Chinese transliterated word pairs from parallel-aligned texts using a statistical machine transliteration model. In *Proceedings of the HLT-NAACL 2003 Workshop on Building and using parallel texts*. Edmonton, Canada, 96–103.

Lee, C.-J., Chang, J. S., and Jang, J.-S. R. 2006a. Alignment of bilingual named entities in parallel corpora using statistical models and multiple knowledge sources. *ACM Transactions on Asian Language Information Processessing 5,* 2, 121–145.

Lee, C.-J., Chang, J. S., and Jang, J.-S. R. 2006b. Extraction of transliteration pairs from parallel corpora using a statistical transliteration model. *Information sciences 176,* 1, 67–90.

Levenshtein, V. I. 1965. Binary codes capable of correcting deletions, insertions and reversals. *Doklady Akademii Nauk SSSR 163,* 4, 845–848.

Li, H., Kuo, J.-S., Su, J., and Lin, C.-L. 2008. Mining live transliterations using incremental learning algorithms. *International Journal of Computer Processing of Oriental Languages 21,* 2, 183–203.

Li, H., Sim, K. C., Kuo, J.-S., and Dong, M. 2007. Semantic transliteration of personal names. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic, 120–127.

Li, H., Zhang, M., and Su, J. 2004. A joint source-channel model for machine transliteration. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona, Spain, 159–166.

Lin, W.-H. and Chen, H.-H. 2002. Backward machine transliteration by learning phonetic similarity. In *Proceeding of the 6th Conference on Natural Language Learning*. Taipei, Taiwan, 1–7.

Lindén, K. 2005. Multilingual modeling of cross-lingual spelling variants. *Information Retrieval 9,* 3, 295–310.

Llitjos, A. F. and Black, A. W. 2001. Knowledge of language origin improves pronunciation accuracy of proper names. In *Proceedings of 7th European Conference on Speech Communication and Technology*. September, 1919–1922.

Loponen, A., Pirkola, A., Järvelin, K., and Keskustalo, H. 2008. A novel implementation of the FITE-TRT translation method. In *30th European Conference on IR Research*. Glasgow, UK, 138–149.

Lu, W.-H., Chien, L.-F., and Lee, H.-J. 2002. Translation of web queries using anchor text mining. *ACM Transactions on Asian Language Information Processing 1,* 2, 159–172.

Malik, M. G. A. 2006. Punjabi machine transliteration. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*. Sydney, Australia, 1137–1144.

Masuyama, T. and Nakagawa, H. 2005. Web-based acquisition of japanese katakana variants. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Salvador, Brazil, 338–344.

Matusov, E., Ueffing, N., and Ney, H. 2006. Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy, 33–40.

McEnery, T. and Wilson, A. 1996. *Corpus Linguistics*. Edinburgh University Press.

Melamed, I. D. 2000. Models of translational equivalence among words. *Computational Linguistics 26,* 2, 221–249.

Meng, H., Lo, W.-K., Chen, B., and Tang, T. 2001. Generate phonetic cognates to handle name entities in English-Chinese cross-language spoken document retrieval. In *Proceedings of the IEEE workshop on Automatic Speech Recognition and Understanding*. Madonna di Campiglio, Italy, 311–314.

Nagata, M., Saito, T., and Suzuki, K. 2001. Using the web as a bilingual dictionary. In *Proceedings of the workshop on Data-driven methods in machine translation*. Toulouse, France, 1–8.

Nomoto, T. 2004. Multi-engine machine translation with voted language model. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. Barcelona, Spain, 494–501.

Nowson, S. and Dale, R. 2007. Charting democracy across parsers. In *Proceedings of the Australasian Language Technology Workshop*. Melbourne, Australia, 75–82.

Och, F. J. and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics 29,* 1, 19–51.

Oh, J.-H. and Choi, K.-S. 2002. An English-Korean transliteration model using pronunciation and contextual rules. In *Proceedings of the 19th International Conference on Computational linguistics*. Taipei, Taiwan.

Oh, J.-H. and Choi, K.-S. 2005. Machine learning based English-to-Korean transliteration using grapheme and phoneme information. *IEICE Transactions on Information and Systems E88-D,* 7, 1737–1748.

Oh, J.-H. and Choi, K.-S. 2006a. An ensemble of transliteration models for information retrieval. *Information Processing and Management 42,* 4, 980–1002.

Oh, J.-H. and Choi, K.-S. 2006b. Recognizing transliteration equivalents for enriching domain-specific thesauri. In *Proceedings of the 3rd International WordNet Conference*. 231–237.

Oh, J.-H., Choi, K.-S., and Isahara, H. 2006a. A hybrid model for extracting transliteration equivalents from parallel corpora. In *Proceedings of 9th International Conference of the Text, Speech and Dialogue*. Brno, Czech Republic, 119–126.

Oh, J.-H., Choi, K.-S., and Isahara, H. 2006b. Improving machine transliteration performance by using multiple transliteration models. In *Proceedings of the 21st International Conference on Computer Processing of Oriental Languages*. Singapore, 85–96.

Oh, J.-H., Choi, K.-S., and Isahara, H. 2006c. A machine transliteration model based on correspondence between graphemes and phonemes. *ACM Transactions on Asian Language Information Processing (TALIP) 5,* 3, 185–208.

Oh, J.-H. and Isahara, H. 2006. Mining the web for transliteration lexicons: Joint-validation approach. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*. Hong Kong, 254–261.

Oh, J.-H. and Isahara, H. 2007a. Machine transliteration using multiple transliteration engines and hypothesis re-ranking. In *Proceedings of the 11th Machine Translation Summit*. Copenhagen, Denmark, 353–360.

Oh, J.-H. and Isahara, H. 2007b. Validating transliteration hypotheses using the web: web counts vs. web mining. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. Silicon Valley, CA, 267–270.

Paczolay, D., Felföldi, L., and Kocsor, A. 2006. Classifier combination schemes in speech impediment therapy systems. *Acta Cybernetica 17,* 2.

Pearson, J. 1998. *Terms in Context*. John Benjamins Publishing Company.

Pedersen, T. 2000. A simple approach to building ensembles of Naïve Bayesian classifiers for word sense disambiguation. In *Proceedings of the 1st Conference on North American Chapter of the Association for Computational Linguistics*. Seattle, Washington, 63–69.

Pervouchine, V., Li, H., and Lin, B. 2009. Transliteration alignment. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Suntec, Singapore, 136–144.

Pirkola, A., Toivonen, J., Keskustalo, H., and Järvelin, K. 2006. FITE-TRT: a high quality translation technique for OOV words. In *Proceedings of the 2006 ACM Symposium on Applied Computing*. Dijon, France, 1043–1049.

Pirkola, A., Toivonen, J., Keskustalo, H., and Järvelin, K. 2007. Frequency-based identification of correct translation equivalents (FITE) obtained through transformation rules. *ACM Transactions on Information Systems 26,* 1.

Rapp, R. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Cambridge, Massachusetts, 320–322.

Rodgers, J. L. and Nicewander, W. A. 1988. Thirteen ways to look at the correlation coefficient. *The American Statistician 42,* 1, 59–66.

Rosti, A.-V., Ayan, N. F., Xiang, B., Matsoukas, S., Schwartz, R., and Dorr, B. 2007. Combining outputs from multiple machine translation systems. In *The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York, 228–235.

Roth, D. and Zelenko, D. 1998. Part of speech tagging using a network of linear separators. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*. Montreal, Canada, 1136–1142.

Sherif, T. and Kondrak, G. 2007a. Bootstrapping a stochastic transducer for Arabic-English transliteration extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Czeck Republic, 864–871.

Sherif, T. and Kondrak, G. 2007b. Substring-based transliteration. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic, 944–951.

Smadja, F. 1992. How to compile a bilingual collocational lexicon automatically. In *Proceedings of the AAAI Workshop on Statistically-Based NLP Techniques*. San Jose, California.

Sproat, R., Tao, T., and Zhai, C. X. 2006. Named entity transliteration with comparable corpora. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the ACL*. Sydney, Australia, 73–80.

Stalls, B. and Knight, K. 1998. Translating names and technical terms in Arabic text. In *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages*. Montreal, Canada, 34–41.

Talvensaari, T., Järvelin, K., and Juhola, M. 2007. Creating and exploiting a comparable corpus in cross-language information retrieval. *ACM Transactions on Information Systems 25,* 1.

Tanaka, K. and Iwasaki, H. 1996. Extraction of lexical translations from non-aligned corpora. In *Proceedings of the 16th Conference on Computational Linguistics*. Copenhagen, Denmark, 580–585.

Tao, T., Yoon, S.-Y., Fister, A., Sproat, R., and Zhai, C. 2006. Unsupervised named entity transliteration using temporal and phonetic correlation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia, 22–23.

Toivonen, J., Pirkola, A., Keskustalo, H., Visala, K., and Järvelin, K. 2005. Translating cross-lingual spelling variants using transformation rules. *Information Processing and Management 41,* 4, 859–872.

Toutanova, K., Ilhan, H. T., and Manning, C. D. 2002. Extensions to HMM-based statistical word alignment models. In *Proceedings of the Conference on Empirical methods in Natural Language Processing*. Pennsylvania, Philadelphia, 87–94.

Tsuji, K., Daille, B., and Kageura, K. 2002. Extracting French-Japanese word pairs from bilingual corpora based on transliteration rules. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation*. Canary Islands, Spain, 499–502.

Van der Eijk, P. 1993. Automating the acquisition of bilingual terminology. In *Proceedings of the 6th Conference of the European Chapter of the Association for Computational Linguistics*. Utrecht, The Netherlands, 113–119.

van Halteren, H., Zavrel, J., and Daelemans, W. 1998. Improving data driven word class tagging by system combination. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*. Montreal, Canada, 491–497.

Virga, P. and Khudanpur, S. 2003a. Transliteration of proper names in cross-language applications. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*. Toronto, Canada, 365–366.

Virga, P. and Khudanpur, S. 2003b. Transliteration of proper names in cross-lingual information retrieval. In *Proceedings of the ACL Workshop on Multilingual and Mixed-Language Named Entity Recognition*. Sapporo, Japan, 57–64.

Vogel, S., Ney, H., and Tillmann, C. 1996. HMM-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational linguistics*. Copenhagen, Denmark, 836–841.

Wan, S. and Verspoor, C. 1998. Automatic English-Chinese name transliteration for development of multilingual resources. In *Proceedings of the 17th International Conference on Computational linguistics*. Montreal, Canada, 1352–1356.

Wu, J.-C. and Chang, J. S. 2007. Learning to find English to Chinese transliterations on the web. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Prague, Czech Republic, 996–1004.

Xu, L., Fujii, A., and Ishikawa, T. 2006. Modeling impression in probabilistic transliteration into chinese. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia, 242–249.

You, J.-L., Chen, Y.-N., Chu, M., Soong, F., and Wang, J.-L. 2008. Identifying language origin of named entity with multiple information sources. *IEEE Transactions on Audio, Speech, and Language Processing 16,* 6, 1077–1086.

Zelenko, D. and Aone, C. 2006. Discriminative methods for transliteration. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia, 612–617.

Zhang, M., Li, H., and Su, J. 2004. Direct orthographical mapping for machine transliteration. In *Proceedings of the 20th International Conference on Computational Linguistics*. Geneva, Switzerland, 716.

Zhang, Y., Huang, F., and Vogel, S. 2005. Mining translations of OOV terms from the web through cross-lingual query expansion. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Salvador, Brazil, 669–670.

Received December 2008; Revised May 2009; July 2009; September 2009; Accepted September 2009.