Inf Technol Manage (2008) 9:91–103
DOI 10.1007/s10799-007-0031-9

Combining probability models and web mining models: a framework for proper name transliteration

Yilu Zhou · Feng Huang · Hsinchun Chen

Published online: 11 December 2007

Abstract The rapid growth of the Internet has created a tremendous number of multilingual resources. However, language boundaries prevent information sharing and discovery across countries. Proper names play an important role in search queries and knowledge discovery. When foreign names are involved, proper names are often translated phonetically which is referred to as *transliteration*. In this research we propose a generic transliteration framework, which incorporates an enhanced Hidden Markov Model (HMM) and a Web mining model. We improved the traditional statistical-based transliteration in three areas: (1) incorporated a simple phonetic transliteration knowledge base; (2) incorporated a bigram and a trigram HMM; (3) incorporated a Web mining model that uses word frequency of occurrence information from the Web. We evaluated the framework on an English–Arabic back transliteration. Experiments showed that when using HMM alone, a combination of the bigram and trigram HMM approach performed the best for English–Arabic transliteration. While the bigram model alone achieved fairly good performance, the trigram model alone did not. The Web mining approach boosted the performance by 79.05%. Overall, our framework achieved a precision of 0.72 when the eight best transliterations were considered. Our results show promise for using transliteration techniques to improve multilingual Web retrieval.

Keywords Name transliteration · Hidden Markov Model · Web mining

1 Introduction

The World Wide Web has become the biggest knowledge repository. There are Web pages in almost every popular language. However, language boundaries prevent information sharing and discovery across countries. There are a wide variety of circumstances in which a reader needs to search for documents in totally unfamiliar languages, for example, companies seeking international business opportunities, researchers seeking references and information on a particular topic, intelligence agencies researching global intelligence, etc.

Proper names, such as organizations, company names, product names, and person names, play an important role in search queries [1]. It was reported that 67.8%, 83.4%, and 38.8% of queries to the Wall Street Journal, Los Angeles Times, and Washington Post respectively involved name searching [2]. In multilingual retrieval most proper names are unknown words that cannot be found in dictionaries, known as out-of-vocabulary (OOV) terms [3]. Those OOV phrases are some of the most difficult phrases to translate because they come from nowhere and are often domain specific [4]. During translation between language pairs employing the same alphabets (e.g., English/Spanish), proper names stay the same. For language pairs employing

Y. Zhou (✉)
Department of Information Systems and Management, George Washington University, Funger Hall, 515N, 2201 G Street, NW, Washington, DC 20052, USA
e-mail: yzhou@gwu.edu

F. Huang
Consumer Electronic Group, Handheld Division, Advanced Micro Devices, Inc., Sunnyvale, USA
e-mail: feng.huang@amd.com

H. Chen
Department of Management Information Systems, The University of Arizona, Tucson, USA
e-mail: hchen@eller.arizona.edu

different alphabets (e.g., English/Arabic), proper names are translated phonetically, referred to as *transliteration*. For example, President "George Bush" is transliterated into Chinese as "乔治 布什" and the company name "SONY" is transliterated into Arabic as "سوني." Being able to identify correct transliterations of proper names as well as identify the origin of transliterated words would largely affect the precision of multilingual Web retrieval and would also be beneficial in machine translation systems or Question Answering systems. While the identification of proper names has received significant attention, transliteration of proper names has not [5].

Recently, transliteration between English and Arabic proper names has drawn much attention. However, automatic transliteration of Arabic names is a challenging task due to the great variation of Arabic language. An Arabic name can have as many as 40 transliterations in English. Even a human translator finds it a difficult task to identify all the variations or recover the Arabic origin from transliterations in this context.

In this research, we aim to develop a generic approach to enable automatic transliteration of Arabic proper names which combines an enhanced Hidden Markov Model (HMM) and a Web mining model. The rest of the paper is structured as follows. Section 1 reviews related research in automatic transliteration and provides a taxonomy of existing approaches. In Sect. 2 we identify research gaps and present our research questions. In Sect. 3 we propose our transliteration framework. Section 4 discusses our experiment design and measures. In Sect. 5 we report and discuss experiment results. Finally, in Sect. 6 we conclude our work and suggest some future directions.

2 Related works

2.1 Transliteration problem

Transliteration is the representation of a word or phrase in the closest corresponding letters or characters of a language with a different alphabet so that the pronunciation is as close as possible to the original word or phrase [6]. It can be classified in two directions: forward transliteration and back transliteration [7]. Consider a name pair (s, t) where s is the original proper name in the source language and t is the transliterated name in the target language. Forward transliteration is the process of phonetically converting s into t. Back transliteration is the process of correctly finding or recovering s given t. Forward transliteration is a one-to-many mapping. For example, the Arabic name "محمد" can be transliterated into "Muhammed," "Mohammed," "Muhamed," etc. Some transliterations might be more popular than others, but it is difficult to define one "correct" transliteration. On the other hand, back transliteration is a many-to-one mapping and has been identified as a more difficult task than forward transliteration [8]. Table 1 classifies previous research with different language pairs according to transliteration directions studied.

Table 1 Transliteration problems studied in previous research

Direction	Forward transliteration	Back transliteration
Process	Phonetically convert to a foreign language	Recover the original name
Feature	One-to-many	Many-to-one
Examples	Clinton → 克林顿	克林顿 → Clinton
	→ 柯林頓	柯林頓
	محمد → Muhammed	Al Qa'ida → قاعِدة
	→ Mohammed	Al Qaeda →
	القاعدة → Al Qa'ida	Al Quieda →
	→ Al Qaeda	Muhammed → مهند
	→ Al Quieda	Mohammed
Previous Research	Arabic → English	Arabic → English
	Arbabi et al. [9]	Stalls and Knight [8]
	English → Arabic	Thai → English
	AbdulJaleel and Larkey [6]	Kawtrakul et al. [13]
	Darwish et al. [10]	Japanese → English
	Al-Onaizan and Knight [5]	Knight and Graehl [14]
	English → Chinese	Goto et al. [15]
	Wan and Verspoor [11]	Chinese → English
	Virga and Khudanpur [12]	Lin and Chen [7]

2.2 Transliteration models overview

Transliteration models can be categorized into four approaches: a rule-based approach, a machine learning approach, a statistical approach, and a Web mining approach.

2.2.1 *Rule-based approach*

A rule-based approach maps each letter or a group of letters in the source language to the closest sounding letter or letters in the target language according to predefined rules or mapping tables. Darwish et al. [10] described a hand-crafted English to Arabic transliteration system. Each English letter was mapped to the closest sounding Arabic letter or letters. All the mapping rules were decided manually. Kawtrakul et al. [13] presented a Thai–English back transliteration using an English phonetic dictionary. Wan and Verspoor [11] described a two-step English to Chinese transliteration, which maps

English into Pinyin and then maps Pinyin into Chinese characters through table lookup.

The rule-based approach is straight forward and easy to implement. It does not rely on any training data. However, it requires manual identification of *all* transliteration rules and heuristics, which is a time-consuming process and sometimes error-prone [10]. Transliteration accuracy depends on the completeness of the rules. Due to the ambiguity of some rules, noise is often introduced. Moreover, this approach is not expandable to different languages pairs.

2.2.2 *Machine learning approach*

The machine learning approach has been adopted in previous research to improve rule-based mapping by filtering out unreliable translations trained from target language patterns. Arbabi et al. [9] used a hybrid neural network and knowledge-based system approach in forward transliteration of Arabic personal names into the Roman alphabet. The neural network was trained on Arabic name samples, and it protects against inaccurate names generated by the rule-based system.

The machine learning approach helps eliminate some ambiguity in transliteration and can be generalized to multiple languages. However, transliteration improvement is often achieved based on a rule-based system. Although some ill-formed transliterations can be removed, it occasionally filters out good transliterations.

2.2.3 *Statistical approach*

A statistical approach is the most promising approach. Instead of relying on a large set of language heuristics, a statistical approach obtains translation probabilities from a training corpus: pairs of transliterated words. This step also requires alignment of training pairs before calculating the probability model. Once the model is trained, on arriving at a new word, the statistical approach picks the transliteration candidate with the highest transliteration probability to generate as the correct transliteration.

Phoneme-based approach: Most previous statistical-based research used phoneme-based transliteration, relying on a pronunciation dictionary. Letter sequences in the source language are first mapped to a phonetic representation acquired from a dictionary, then mapped to letter sequences in the target language. Knight and Graehl [14] described a phoneme-based probabilistic model for an English–Japanese back-transliteration system. Their probability model first transformed written English into English pronunciation, then to a Japanese sound inventory, and finally into written Japanese words (katakana). Using a similar approach, Stalls and Knight [8] developed a probabilistic model of English Arabic transliteration. The phoneme-based approach fails when such a dictionary is not available. Meng et al. [16] reported 47.5% syllable accuracy during English–Chinese transliteration where 2,233 name pairs were used as the training corpus. More recently Virga and Khudanpur [12] relied on a text-to-speech system to obtain phonemic pronunciation of each English name in English–Chinese name transliteration. Their training sample size was the same as in Meng's work.

Phoneme-based mapping is quite effective when a pronunciation dictionary is available. It handles multi-letter combinations successfully. However, only words with known pronunciation can be produced and it cannot deal with OOV terms. It could fail in back transliteration, since many foreign names, such as Muhammed, are not likely to be in a dictionary [5].

Grapheme-based approach: The grapheme-based approach uses probability to directly maps letter sequences in a source language into letter sequences in the target language. This approach is often used for transliterations between two alphabet-based languages, such as English/Arabic, English/Russian, etc. Al-Onaizan and Knight [5], in a study involving Arabic–English transliteration, showed that a grapheme-based model achieved better accuracy than a state-of-the-art phoneme-based model, and the mixed phoneme- and grapheme-based approach only slightly improved the accuracy over the grapheme-based approach. To filter out ill-formed name strings, they added a Web-based filtering step which eliminated candidates with zero Web counts. However, their transliteration model did not consider the context information of alphabets, which could harm performance. AbdulJaleel and Larkey [6] also presented a grapheme-based statistical method for English to Arabic forward transliteration. They concluded that a bigram model outperformed a unigram model in English Arabic transliteration, because the bigram model considers the context to some degree. They used 5,000, 10,000 and 50,000 name pairs respectively as training data, and reached 43.4% accuracy with a training sample of 50,000. But no significant differences were found with varied training sample sizes.

Unlike the phoneme-based approach, the grapheme-based approach does not require a phonetic dictionary or linguistic rules. However, it is likely that a given letter sequence in a source language might generate an ill-formed phoneme sequence in a target language in a solely grapheme-based mapping.

2.2.4 *Web mining-based approach*

The Web mining-based approach takes a very different view of the transliteration problem. Web mining is defined as the discovery and analysis of useful information from the WWW. It can be categorized into Web content mining, Web structure mining and Web usage mining [17]. Web content mining deals with web contents/data/documents/services. Structure mining copes with hyperlinks between websites. Web usage mining utilizes data generated by users' interaction with the Web, such as server logs and user profiles. Unlike rule-based approach, Web mining-based approach does not rely on transliteration heuristics or probability models. Instead, it searches the Web for transliteration using relevant context words of the source name. The assumption here was that the two name equivalents should share similar relevant context words in their languages. Correct transliteration is then extracted from the closest matching proper nouns.

Goto et al. [15] proposed such an Internet-based technique for finding English equivalents for Japanese names. They first searched the Internet for relevant context words of the original name, and then used the translated context words as a query to obtain relevant Web documents. Similarly, Lu et al. [18] presented an approach to finding translation equivalents of query terms and constructing multilingual lexicons through the mining of Web anchor texts and link structures, which was shown to be effective on English–Chinese Web documents.

The Web mining approach is applicable to any pairs of languages. No rules, dictionaries, or training corpora are needed. However, the performance depends on the ability to identify proper names and accuracy in translating relevant context words. This approach works well for hotspots in news articles, but not normal names.

2.3 A taxonomy of transliteration research

Proper name transliteration is an important problem in many applications which has not been widely studied. We present a taxonomy of transliteration approaches in Tables 2 and 3. Table 2 describes major transliteration models and Table 3 further illustrates different approaches in the statistical model.

3 Research questions

Based on our review, several research gaps have been identified. Statistical approaches are the most promising, but little of the research has considered context information in the transliteration model. Although Al-Onaizan and Knight [5] used Web counts to filter out unreliable

Table 2 Taxonomy of transliteration research

Models	Resources	Descriptions	Examples
Rule-based	Mapping heuristics and knowledge	Transliteration is based on heuristics of source and target languages	Darwish et al. [10] Wan and Verspoor [11] Kawtrakul et al. [13]
Machine learning enhanced	Training samples of words in target language	Machine learning algorithms such as Neural Network are used to filter out ill-formed transliterations	Arbabi et al. [9]
Statistical approach	Training samples (list of transliteration pairs)	Translation probabilities are learned from a training sample of transliterated words in two languages	See Table 3
Web mining approach	Comparable Web context of proper names in both languages	Extract proper names from relevant context in both languages, and then compare their pronunciation similarity to match transliterations	Goto et al. [15] Lu et al. [18]

Table 3 Taxonomy of transliteration research using statistical approach

Models	Resources	Descriptions	Examples
Statistical approach			
Grapheme-based	Pairs of transliteration samples	Directly maps letter sequences in source language into letter sequences in target language	AbdulJaleel and Larkey [6] Al-Onaizan and Knight [5]
Phoneme-based	Phonetic dictionary; Pairs of training samples	Letter sequences in source language are mapped to their phonemic representations acquired from a dictionary first, and then mapped to letter sequences in target language	Virga and Khudanpur [12] Knight and Graehl [14] Stalls and Knight [8] Meng et al. [16]

transliteration, it remains unknown how and to what extent a Web mining model could enhance the probability model. It is a challenge to develop a generic approach for name transliteration to support knowledge discovery in multilingual content. We propose the following research questions.

1. How can we build a statistical *transliteration model* that does not rely on human-defined rules?
2. How can we utilize *context information* in transliteration?
3. How can we integrate a *Web mining* component to improve system performance?
4. Does this model achieve satisfactory *effectiveness*?

4 Proposed framework

Aiming to develop a generic framework with less human intervention and more easily obtained resources, we propose to adopt a grapheme-based statistical approach in proper name transliteration. Most previous research used a simple statistical approach with independent probability estimation, assuming that transliteration of letters is context-independent. Correct transliteration is dependent on both source and target word context. We propose to use the HMM, which is one of the most popular probability models and has been used in speech recognition, the human genome project, consumer decision modeling, etc. [19], yet has seldom been explored in proper name transliteration. HMM fits the transliteration problem well. Since the model translates the current grapheme based on the observation of the previous grapheme transliterated, it captures context information. Furthermore, by examining the popularity of all possible transliterations on the Internet, bad transliterations can be filtered and their online popularity can serve as an indicator of transliteration correctness.

The proposed framework makes improvements in three aspects: (1) incorporating a simple phonetic transliteration knowledge base, (2) incorporating a bigram and a trigram HMM, and (3) incorporating a Web mining model to identify the most popular transliteration. It is composed of a training process and a transliteration process as shown in Figs. 1 and 2. We explain the detailed components in each process in Sects. 3.1 and 3.2.

4.1 Training statistical model

The *training process* generates transliteration probabilities based on a training corpus (Fig. 1). There are three steps in the training process: (1) to Construct a Knowledge Base, (2) to Align Pairs of Transliterations, and (3) to Generate the Statistical Model.

4.1.1 Phonetic knowledge base

The first step in training is to *Construct a Simple Phonetic Knowledge Base* (KB), which consists of general phonetic rules for name parsing and alignment. In this step, multi-letter phonemes are identified as one transliteration unit. For example, “ou,” “th,” and “ee” are multi-letter phonemes in English. Restriction rules for alignment are also identified. For example, the English letter “a” can map to “ا“ ,”ع“ ,”ة”, or “ى” in Arabic. This is the component where some language specific features are captured. Note that the

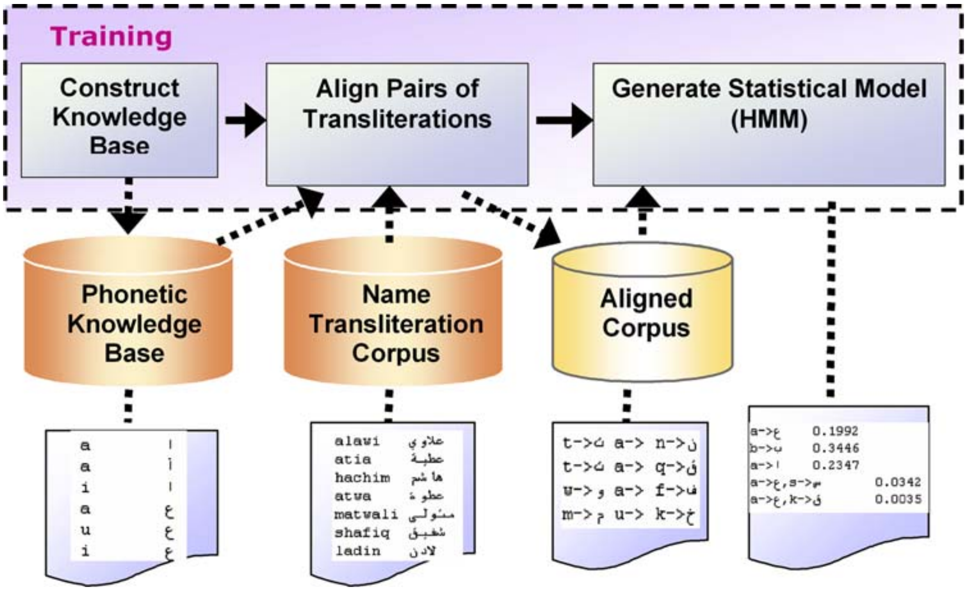


Fig. 1 Training statistical model

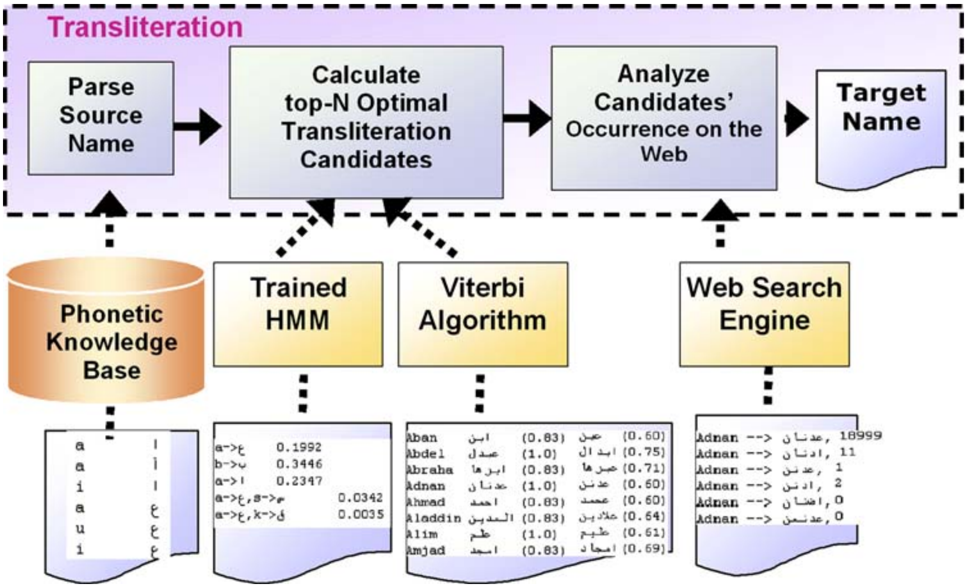

Transliteration
Parse Source Name
Calculate top-N Optimal Transliteration Candidates
Analyze Candidates' Occurrence on the Web
Target Name
Phonetic Knowledge Base
Trained HMM
Viterbi Algorithm
Web Search Engine
a ا
a آ
i ا
a ع
u ع
i ع
a->ع 0.1992
b->ب 0.3446
a->ا 0.2347
a->ع,s->ص 0.0342
a->ع,k->ق 0.0035
Aban ابن (0.83) عبن (0.60)
Abdel عبدل (1.0) ابدال (0.75)
Abraha ابرها (0.83) عبرها (0.71)
Adnan عدنان (1.0) عدنن (0.60)
Ahmad احمد (0.83) عحمد (0.60)
Aladdin الددين (0.83) علادين (0.64)
Alim علم (1.0) طبع (0.61)
Amjad امجد (0.83) امجاد (0.69)
Adnan --> عدنان, 18999
Adnan --> ادنان, 11
Adnan --> عدنن, 1
Adnan --> ادنن, 2
Adnan --> اضنان, 0
Adnan --> عدنمن, 0

Fig. 2 Transliteration process

knowledge base is much less complex than that used in a rule-based system.

Table 4 gives an example of a phonetic knowledge base between English letters and Arabic letters. This knowledge base consists of three types of mappings: single letter, multiple letters and special vowels. The first three columns are mapping from single Roman letters to Arabic letters which are most common. The fourth column represents mapping between multiple Roman letters and single Arabic letters, some of which are long vowels and special consonants in Arabic. The last column maps weak vowels that are sometimes omitted in written Arabic names.

4.1.2 Alignment

The second step is to *Align Pairs of Transliterations*. *Alignment* is a process that connects each letter or transliteration unit in the source language with a letter or transliteration unit in the target language. Different text alignment approaches have been proposed in Machine Translation and Cross-lingual Information Retrieval research, such as Finite State Automata (FSA), backtracking methods, the EM-algorithm, etc. Most of them deal with complex linguistic context. Since word context is less complex than that of texts, we use a simple and efficient left-to-right, one-step-backtracking method to produce optimal alignment.

The alignment step starts from the first letter (or letter group) in the source name and assumes a mapping with the first letter (or letter group) in the target name if no restrictions are found in the KB. If KB violations are found, the program either jumps to the second letter in the target name for a potential mapping with the first letter in the source name, or it jumps to the second letter in the source name for a potential mapping with the first letter in the target name, and so on. An unmapped letter in the target name is considered to be an omitted pronunciation during transliteration, and an unmapped letter in the source name is considered to be an over-generalized pronunciation during transliteration.

Table 4 An example of English–Arabic phonetic knowledge base

Single letter						Multiple letters		Weak vowels	
'	ع	h	ة	r	ر	aa	ا	e	´
a	ا	h	ح	r	غ	aa	ع	a	´
a	آ	h	ع	s	ز	au	و	i	´
a	ة	h	ه	s	س	ch	ش	y	´
a	ع	i	ئ	s	ص	dh	ذ	u	‚
a	ى	i	ا	t	ت	dh	ض	ou	‚
b	ب	i	ع	t	ث	dh	ظ	o	‚
c	س	i	ي	t	ط	dj	ج		
c	ك	j	ج	u	ا	ee	ي		
d	د	j	خ	u	ع	ge	ج		
d	ذ	k	خ	u	و	gh	غ		
d	ض	k	ق	v	ف	kh	خ		
e	ئ	k	ك	w	و	oo	و		
e	ا	l	ل	y	ي	ou	و		
e	ي	m	م	z	ذ	sh	ش		
f	ف	n	ن	z	ز	th	ث		
g	ج	o	ع	z	ظ	th	ذ		
g	غ	o	و			uw	و		
g	ق	q	ق						

Examples of English–Arabic transliteration alignment are shown in Table 5. The inputs are English and Arabic name pairs and the outputs are mappings of letters identified in the name pairs. The special character "@" represents unmapped letters.

4.1.3 Statistical model

The last step in the training process is to *Generate the Statistical Model*, or the probability model. The model is derived from frequency counts of letter mappings observed in the aligned training corpus. Most previous research used a simple statistical model with independent probability estimation and we use this approach as our benchmark. We also investigate three more advanced statistical models: a bigram HMM, a trigram HMM, and a combination of bigram and trigram HMM.

All statistical models try to find the candidate transliteration with the highest transliteration probabilities:

$$\arg\max P(t|s) = \arg\max P(t_1 t_2 \ldots t_n | s_1 s_2 \ldots s_m) \quad (1)$$

where $P(t_i|s_i) = \frac{\text{No. of times } S_i \text{ translates to } \mathrm{t_i}}{\text{No. of times } S_i \text{ occurs}}$, and

$$P(t_i|s_i, t_{i-1}) = \frac{\text{No. of times } s_i \text{ translates to } t_i \text{ given } s_{i-1} \rightarrow t_{i-1}}{\text{No. of times } s_{i-1} \text{ translates to } t_{i-1}}$$

In some cases, the translation probability of the current letter depends not only on one state before the current state (or one letter/transliteration unit before the current character), but on two or more states. The *trigram HMM* intends to capture even more context information by translating the current letter dependent on the *TWO* previous letters. Transliteration probability is estimated as:

$$P(t_1, t_2, t_3, \ldots, t_n | s_1, s_2, s_3, \ldots, s_n) \\ = P(t_1|s_1) p(t_2|s_2, t_1) P(t_3|s_3, t_2, t_1) \ldots p(t_n|s_n, t_{n-1}, t_{n-2}) \quad (4)$$

where

$$P(t_i|s_i) = \frac{\text{No. of times } s_i \text{ translates to } t_i}{\text{No. of times } s_i \text{ occurs}},$$

$$P(t_i|s_i, t_{i-1}) = \frac{\text{No. of times } s_i \text{ translates to } t_i \text{ given } s_{i-1} \rightarrow t_{i-1}}{\text{No. of times } s_{i-1} \text{ translates to } t_{i-1}}$$

and

$$P(t_i|s_i, t_{i-1}, t_{i-2}) = \frac{\text{No. of times } s_3 \text{ translates to } t_i \text{ given } s_{i-1} \rightarrow t_{i-1} \text{ and } s_{i-2} \rightarrow t_{i-2}}{\text{No. of times } s_{i-1} \text{ translates to } t_{i-1} \text{and } s_{i-2} \text{ translates to } t_{i-2}}$$

where s is the source name to be transliterated, which contains letter string $s_1 s_2 \ldots s_i$; t is the target name, which contains letter string $t_1 t_2 \ldots t_i$.

In a *simple statistical model*, transliteration probability is estimated as:

$$P(t_1, t_2, t_3, \ldots, t_n | s_1, s_2, s_3, \ldots, s_n) = P(t_1|s_1) P(t_2|s_2) \ldots P(t_n|s_n) \quad (2)$$

where

$$P(t_i|s_i) = \frac{\text{No. of times } s_i \text{ translates to } \mathrm{t_i} \text{ in corpus}}{\text{No. of times } s_i \text{ appears in corpus}}$$

The *bigram HMM* improves the simple statistical model in that it incorporates context information into a probability calculation. The transliteration of the current letter is dependent on the transliteration of *ONE* previous letter (one previous state in HMM). Transliteration probability is estimated as:

$$P(t_1, t_2, t_3, \ldots, t_n | s_1, s_2, s_3, \ldots, s_n) \\ = P(t_1|s_1) P(t_2|s_2, t_1)(t_3|s_3, t_2) \ldots p(t_n|s_n, t_{n-1}) \quad (3)$$

The *combined bigram and trigram model* estimates the weighed probability.

We give an example of transliteration probability under different conditions in Table 6. The first column shows the transliteration probability of letter "a" under a simple statistical model. The second column again shows the transliteration probability of letter "a" under the condition that the previous letter is "s" and "s" is transliterated into "س." This conditional probability is used in bigram HMM. The third column is the transliteration probability of "a" under a stronger condition, where the previous two letters are "l" and "s" and are transliterated into "ل" and "س" respectively. This probability is used in trigram model. Note that when the condition becomes stronger, there are higher chances that such condition does not exist in training data. For example, there is no observation of a → ة when the previous transliteration is s → س. Considering that the training data might not be comprehensive enough to cover all conditions, we use ε, a very small number to represent those conditional probabilities that are not observed in the training data.

Table 5 Examples of English–Arabic transliteration alignment

Inputs	Outputs
abas عباس	a → ع b → ب a → ا s → س
abou ابو	a → ع b → ب ou → و
abubakar ابوبَكار	a → ع b → ب u → ا b → ب a → ع k → ك a → ´ r → @
ademi ادِمي	a → ا d → ض e → ِ m → @ i → ِ
hamed حامِد	h → ه a → ا m → م e → ا d → د
kasim قاسِم	k → ق a → ا s → س i → ي m → م
omran عمران	o → ع m → م r → غ a → ´ n → @
rahim رَحيم	r → ر a → ع h → ه i → ِ m → @
saeed سعيد	s → ص a → ى ee → ي d → ض
sayaf سَياف	s → ص a → ى y → ي a → ´ f → @

4.2 Transliteration process

The *transliteration process* transliterates proper names using the probability model obtained from the training process (Fig. 2). It contains three steps: (1) to Parse Source Names, (2) to Generate Top-N Transliteration Candidates, and (3) to Analyze Candidates' Occurrences on the Web (Web mining approach).

4.2.1 Source name parsing

The first step in the transliteration process is to *Parse Source Names*. Source names are first tokenized against letters or multi-letter phonemes identified in the Phonetic Knowledge Base. These tokenized units, most of which are single letters, are used as input for the statistical model. For example, "Ghunaym" is parsed into {gh, u, n, a, y, m} and "Ishaq" is parsed into {i, s, h, a, q}.

4.2.2 Top-N optimal transliterations

The next step is to *Calculate Top-N Optimal Transliteration Candidates* based on trained probabilities. When feeding the model with a new proper name in the source language, the most probable transliteration is a letter sequence path that maximizes *P(t|s)*. As we described in Sect. 3.1, *P(t|s)* is evaluated as a sequence of consecutive letter mappings and the conditional probability of each letter mapping can be estimated from the training corpus. In other words, *P(t|s)* is calculated as the multiplication of sequences of conditional probabilities according to the statistical model used.

Table 6 An example of transliteration probability table

P(a → *)		P(a → *|s → س)		P(a → *|(l → ل) and (s → س))	
a → ع	0.081954	a → ع	0.102041	a → ع	ε
a → ا	0.411347	a → ا	0.204082	a → ا	0.714286
a → ´	0.431836	a → ´	0.632653	a → ´	0.285714
a → ة	0.057525	a → ة	ε	a → ة	ε
a → ى	0.015760	a → ى	0.061224	a → ى	ε
a → آ	0.001576	a → آ	ε	a → آ	ε

However, calculating all the possible sequences with such a large number of parameters is overwhelming. Thus, we use Viterbi's search algorithm for finding the most likely sequence of target transliteration letters that result in a sequence of source names. Viterbi's algorithm is a dynamic programming algorithm which is often used in the context of HMM [20]. Instead of keeping one optimal path, we keep the top-N optimal paths as our transliteration candidates.

For example, we want to back transliterate "Ishaq" into its Arabic origin. Figure 3 illustrates all the possible paths that the transliteration could be performed. The weight of each path is the probability estimated from equations in Sect. 3.1.3. Using Viterbi algorithm, we then identifies the top-N optimal paths (paths with the highest weights).

4.2.3 Web occurrence analysis

To boost the transliteration performance we propose to use the Web mining approach, which *Analyzes Candidates' Occurrence on the Web***.** Each one of the top-N transliterations obtained from the previous step is sent to a Web search engine using a meta-search program which records the number of documents retrieved, referred to as Web frequency. This information is an indicator of the candidate

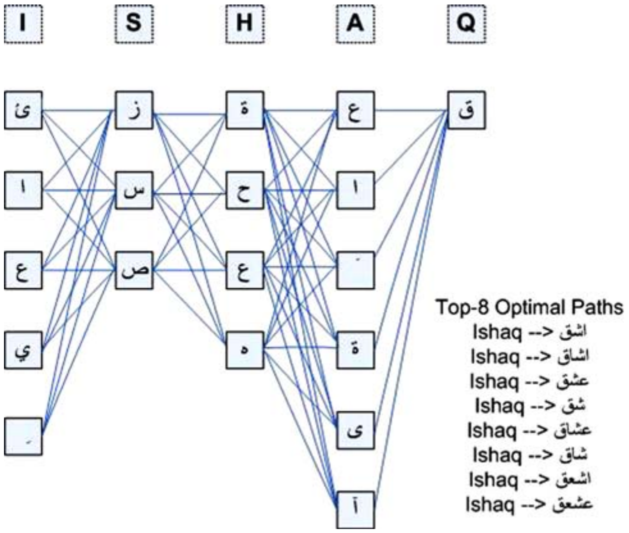


Fig. 3 Transliterating "Ishaq" into its Arabic origin

Table 7 An example of web occurrence analysis

Name in roman letters	Transliteration candidates	MSN search results (# of Web pages that contains the transliteration candidate)
Ishaq	اشق	145
	اشاق	2
	عشق	126,231
	شق	9,185
	عشاق	15,604
	شاق	1,968
	اشعق	0
	عشعق	0

transliteration's online popularity. The more often the candidate transliteration appears in online documents, the more likely it is a correct transliteration.

Table 7 takes "Ishaq" as an example. The eight Arabic origins generated from the statistical model in Sect. 3.2.2 are listed in the second column. However, the Arabic candidates' online occurrences from MSN search (search.msn.com) are very different. The third best transliteration from the statistical model turns out to have the most occurrences on MSN search: appears in 126,231 Web pages.

Unlike Al-Onaizan and Knight's work [5], we do not throw away candidates with zero Web counts. Both Web frequency information and transliteration probability of top-N candidates contribute to the final score formula that is used to rank transliteration candidates:

$$\begin{aligned}\text{Final score} = {} & \alpha * \text{normalized probability score} \\ & + \beta * \text{normalized Web frequency}, \\ & \text{s.t. } \alpha + \beta = 1. \end{aligned} \tag{5}$$

This final rank of transliterations is derived from a weighed score of the normalized Web frequency and the probability score. On the one hand, even though we are using Web mining for disambiguation, we do not want to treat all the top-N transliteration candidates equally. Instead, we retain information from the probability model. In this way, if two transliterations have a similar Web frequency score (e.g., 128,000 vs. 128,001) their probability scores will play a major role in selecting the best transliteration. On the other hand, we still want to distinguish between different Web frequency counts if the difference is big enough. In transliteration the occurrence difference between 128,000 and 1 should have a much bigger effect than the difference between 128,000 and 127,000, in which case the Web frequency score will play a more important role in the final ranking score. In our framework, we used linear normalization. During each transliteration, Web frequency and probability score for each candidate were divided by the highest ones achieved among all candidates. We chose α = 0.5 and β = 0.5 to generate the final score. This setting gave same weights to the probability model and the Web mining model. Other settings of α and β were not tested in this work. We have interest in testing the effect of different α and β settings in the future. All the transliteration candidates are then ranked by their final scores.

5 Experimental design

We designed experiments to study the performance of our proposed research framework using different statistical models and using a Web mining model. In this section we present the hypotheses and experimental design.

5.1 Hypotheses

We are interested in the performance of eight experimental settings: (1) A simple statistical approach, (2) A bigram HMM approach, (3) A trigram HMM approach, (4) A hybrid HMM approach (bigram + trigram), and each of the above condition with a Web-mining-enhanced approach.

In H1.1–H1.3, we studied the performance of the probability model alone. A simple statistical approach has been adopted in previous transliteration research, and we used it as our benchmark. A bigram HMM is a traditional HMM, which predicts the grapheme transliteration based on the conditional probability of one previous grapheme transliteration observed in training data. We believe that incorporating the HMM would improve performance. A trigram HMM is an improved HMM which integrates a more complex conditional probability model and captures two previous grapheme transliterations. It provides a stronger relation between word graphemes. Furthermore, we believe that a combined bigram and trigram model could complement each other and further improve the performance over a trigram model alone. Thus, we hypothesized that:

H1.1: A bigram HMM approach performs better than a simple statistical approach.

H1.2: A trigram HMM approach performs better than a bigram HMM approach.

H1.3: A hybrid HMM approach performs better than a trigram HMM or a bigram HMM alone.

In H2.1–2.4, we looked at the effect of integrating a Web mining model with probability models. A Web mining model provides additional information on transliterations' online popularity. We believed that a combined model would always outperform a single probability model. Thus, we hypothesized that:

H2.1: Integrating a Web mining model improves a simple statistical approach significantly.

H2.2: Integrating a Web mining model improves a bigram HMM approach significantly.

H2.3: Integrating a Web mining model improves a trigram HMM approach significantly.

H2.4: Integrating a Web mining model improves a hybrid HMM approach significantly.

5.2 Measures

Previous transliteration research has used "accuracy" to measure performance, which is defined as:

$$\text{Accuray} = \frac{\text{Number of correct transliterations}}{\text{Total number of transliterations}} \tag{6}$$

Besides measuring the accuracy for the highest ranked transliteration, identifying a set of optimal transliteration candidates is of interest. Top-*N* accuracy is often used in translation and transliteration research. It is defined as the percentage of names whose selected top-*N* transliterations include correct transliterations:

$$\text{Top-N accuracy} = \frac{\text{Total number of times correct transliterations appeared in the first } N \text{ candidates}}{\text{Total number of transliterations performed}} \tag{7}$$

The traditional accuracy measure can be viewed as top-1 accuracy. Only when the first selected candidate is the correct transliteration, will it be considered as a hit. For top-2 accuracy, when one of the top-2 candidates is a correct transliteration, it will be considered as a hit. In our experiments, we chose *N* = 1, 2, 4, 8.

5.3 Dataset

Our English–Arabic transliteration dataset is a list of 1,000 unique Arabic names extracted from http://www.ummah.net/family/masc.html. An Arabic-speaking expert manually translated all Arabic names into English transliterations according to his knowledge. Because we focus on back transliteration from English to Arabic, each English transliteration should be converted to one and only one correct Arabic name. This dataset is unaligned.

5.4 Experiment methodology

We used the 10-fold cross validation method, commonly used in testing data mining algorithms and models, to test system accuracy. We first randomly divided the data into 10 subsets of equal size. We trained the model 10 times, each time leaving out one of the subsets, to compute the system's top-N accuracy. Accuracy scores obtained from each subset were then averaged.

6 Experiment results and discussion

In this section we describe and analyze the results of our experiments. Table 8 presents the overall transliteration performance results (measured by top-N accuracy) under five experiment conditions and their improvement over a simple statistical model. The best performance was achieved using a combined hybrid HMM and Web mining model (column 5), a 0.38 top-1 accuracy and a 0.72 top-8 accuracy. Bigram HMM, hybrid HMM, and a combined hybrid HMM and Web mining model enhanced a simple statistical approach by 20.87%, 62.09%, and 79.05%, respectively for top-1 accuracy. Surprisingly, trigram HMM degraded the performance by 81.59%. Improvements in the top-2, top-4, and top-8 accuracy were not as tremendous as that of the top-1, ranging from 3.09% (top-4 accuracy for bigram) to 25.88% (top-2 accuracy for Web mining enhanced).

6.1 Comparison of probability models

Table 9 reports average accuracy achieved from four different probability models and our paired *t*-test results. Figure 4 illustrates the differences among all four approaches.

The results for our hypotheses showed that H1.1 and H1.3 were supported, but H1.2 was not. There were significant improvements from the simple statistical approach to the bigram HMM approach. However, the performance significantly decreased when using trigram HMM alone. There are two possible causes for the drop in accuracy. First, we believe that, overall, bigram HMM is a better model for English–Arabic transliteration. Most Arabic name transliteration processes depend on just one letter ahead of the current one, instead of two letters ahead. Second, we observed that because trigram HMM is a strong

Table 8 Summary of system performance (accuracy) with different models and their improvement over a Simple Statistical model

	Simple	Bigram	Impr. over simple (%)	Trigram	Impr. over simple (%)	Hybrid	Impr. over simple (%)	Web mining enhanced	Impr. over simple (%)
Top-1	0.21	0.26	20.87	0.04	81.59	0.34	62.09	0.38	79.05
Top-2	0.41	0.44	5.20	0.08	79.84	0.50	20.54	0.52	25.88
Top-4	0.57	0.59	3.09	0.12	78.58	0.63	9.94	0.64	11.34
Top-8	0.66	0.69	3.88	0.22	66.72	0.72	8.26	0.72	8.26

Simple: simple statistical model

Bigram: bigram HMM

Trigram: trigram HMM

Hybrid: Hybrid HMM (bigram + trigram)

Web-mining-enhanced: Hybrid HMM + Web mining model

Impr. over simple: Improvement achieved over simple statistical model

Table 9 Summary of average accuracy achieved and *t*-test results

	Simple		Bigram		Trigram		Hybrid	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Average accuracy achieved (Probability models)								
Top-1	.21	0.03	0.26	0.05	0.04	0.03	0.34	0.05
Top-2	0.41	0.03	0.44	0.05	0.08	0.04	0.50	0.05
Top-4	0.57	0.02	0.59	0.03	0.12	0.03	0.63	0.04
Top-8	0.66	0.03	0.69	0.04	0.22	0.05	0.72	0.05
Paired t-test (2 tail, α = 0.05)								
P_{simple}			1.09E-05		2.2E-20		2.13E-11	
P_{bigram}					1.29E-22		1.14E-10	
P_{trigram}							9.62E-28	

relation, it needs a large training dataset to obtain all possible triple-letter sequences. Our training data of 900 Arabic names might not be sufficient. The hybrid model performed significantly better than a bigram model, which implied that when trigram probability existed in training data it helped improve the performance. We concluded that a hybrid HMM which combines bigram and trigram information yielded the best performance in our experiments and the top-8 accuracy reached 0.72 with a relatively small training dataset.

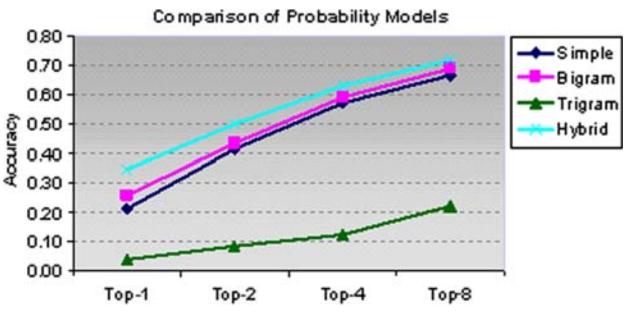


Fig. 4 Performance comparison of probability models (accuracy)

6.2 Performance of web mining model

In Table 10, we report top-N accuracy achieved with the four probability models and their corresponding Web-mining-enhanced models. We provide our paired *t*-test results of comparing the probability model alone and a combined probability and Web mining model in the same table. Figure 5 illustrates the improvements obtained from the Web mining model.

As we hypothesized in H2.1 to H2.4, Web mining always advanced the performance of the probability model significantly, no matter which probability model we used. H2.1–H2.4 were all supported. This confirmed that online occurrence information obtained from search engines is an effective way to identify the correct transliterations. There is a pattern of larger enhancement on lower accuracy, and smaller enhancement on higher accuracy. The improvements achieved on top-1 and top-2 accuracy were more obvious than that obtained on top-4 and top-8 accuracy. Similarly, the boosting effects on simple, bigram, and trigram models were more noticeable than that in a hybrid model.

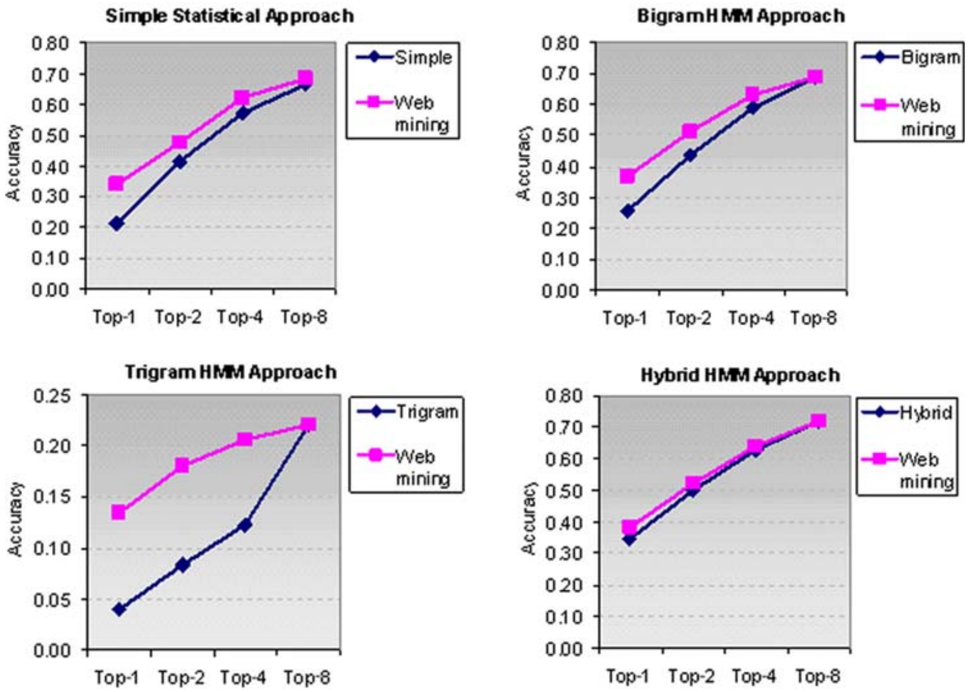
7 Conclusions and future directions

In this research we proposed a generic proper name transliteration framework which incorporated the HMM and Web mining approaches. We evaluated the framework with English–Arabic back transliteration. We found that a bigram HMM significantly improved the performance over a simple statistical approach. While a trigram HMM did not improve the accuracy, a combination of a bigram and a trigram HMM method outperformed a bigram HMM alone. We believe that in the hybrid HMM approach, the bigram HMM and the trigram HMM complement each other and thus yield the best performance among all four probability models tested. The top-1 accuracy reached 0.34 and top-8

Table 10 Summary of average accuracy achieved and *t*-test results using combined Probability and Web mining model

	Simple		Simple + Web mining		Bigram		Bigram + Web mining		Trigram		Trigram + Web mining		Hybrid		Hybrid + Web mining	
	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.	Avg.	Std.
Summary of results (Probability + Web mining models)																
Top-1	0.21	0.03	0.34	0.05	0.26	0.03	0.37	0.03	0.04	0.02	0.13	0.05	0.34	0.03	0.38	0.03
Top-2	0.41	0.04	0.48	0.05	0.44	0.05	0.51	0.03	0.08	0.03	0.18	0.05	0.50	0.04	0.52	0.04
Top-4	0.57	0.03	0.62	0.05	0.59	0.04	0.63	0.05	0.12	0.03	0.21	0.05	0.63	0.05	0.64	0.05
Top-8	0.66	0.05	0.68	0.05	0.69	0.05	0.69	0.05	0.22	0.04	0.22	0.04	0.72	0.05	0.72	0.05
Paired t-test (2 tail, α = 0.05)																
P value			H2.1 2.44E-08				H2.2 2.76E-08				H2.3 2.76E-08				H2.4 0.0001	

Fig. 5 Performance comparison of combined probability and Web mining models (accuracy)



accuracy reached 0.72. The Web mining approach boosted the performance by 79.05% for top-1 and 8.26% for top-8 over the simple statistical model. The boosting effect for hybrid HMM is not as great as that for the simple statistical model, and the effect is larger for top-1 and top-2 accuracy compared to top-4 and top-8 accuracy. Frequency information obtained from the Web proved an effective way to identify the correct transliteration. Overall, the results are very promising.

Our framework of transliteration has several practical applications. For example, it could improve the performance of current multilingual Web retrieval by transliterating out-of-vocabulary proper nouns. It could also be adopted in machine translation systems. In the future, we plan to test our framework on more language pairs and incorporate a transliteration component into multilingual Web retrieval systems.

References

1. G.-W. Bian, H.-H. Chen, Cross-language information access to multilingual collections on the internet J. Am. Soc. Inf. Sci. **51**, 281 (2000)
2. P. Thompson, C.C. Dozier, Name Searching and Information Retrieval, in *Proceedings of 2nd Conference on Empirical Methods in Natural Language Processing* (Providence, Rhode Island, 1997)
3. H.-H. Chen, S.-J. Hueng, Y.-W. Ding et al., Proper name translation in cross-language information retrieval, in *Proceedings of the 17th International Conference on Computational Linguistics* (Montreal, 1998), p. 232
4. Y. Al-Onaizan, K. Knight, Translating named entities using monolingual and bilingual resources, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (2001), p. 400
5. Y. Al-Onaizan, K. Knight, Machine transliteration of names in Arabic text, in *Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages* (Philadelphia, 2002), pp. 1

6. N. AbdulJaleel, L.S. Larkey, Statistical transliteration for English–Arabic cross language information retrieval, in *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM)* (New Orleans, 2003), p. 139
7. W.-H. Lin, H.-H. Chen, Backward machine transliteration by learning phonetic similarity, in *Proceedings of The 6th Workshop on Computational Language Learning (CoNLL-2002)* (Taipei, 2002), p. 139
8. B.G. Stalls, K. Knight, Translating names and technical terms in Arabic text, in *Proceedings of the COLING/ACL Workshop on Computational Approaches to Semitic Languages* (Montreal, 1998)
9. M. Arbabi, S.M. Fischthal, V.C. Cheng, et al., Algorithms for Arabic Name Transliteration. IBM J. Res. Dev. **38**, 183 (1994)
10. K. Darwish, D. Doermann, R. Jones, et al., TREC-10 experiments at University of Maryland CLIR and video, in *Text REtrieval Conference* (Gaithersburg, 2001)
11. S. Wan, C.M. Verspoor, Automatic English–Chinese name transliteration for development of multilingual resources, in *Proceedings of the 17th international conference on Computational linguistics* (Montreal, 1998), p. 1352
12. P. Virga, S. Khudanpur, Transliteration of proper names in cross-lingual information retrieval, in *Proceedings of the ACL Workshop on Multi-lingual Named Entity Recognition* (Sapporo, 2003), p. 57
13. A. Kawtrakul, A. Deemagarn, C. Thumkanon, et al., Backward transliteration for Thai document retrieval, in *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAD)* (Chiangmai, 1998), p. 563
14. K. Knight, J. Graehl, Machine transliteration, in *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics* (Somerset, 1997), p. 128
15. I. Goto, N. Uratani, T. Ehara, Cross-language information retrieval of proper nouns using context information, in *Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium* (Tokyo, 2001), p. 571
16. H. Meng, W.-K. Lo, B. Chen, et al., Generating phonetic cognates to handle named entities in English–Chinese cross-language spoken document retrieval, in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding Workshop (ASRU)* (Trento, 2001), p. 311
17. S.K. Pal, V. Talwar, P. Mitra, Web mining in soft computing framework: relevance, state of the art and future directions. IEEE Trans. Neural Networks **13**, 1163 (2002)
18. W.-H. Lu, L.-F. Chien, H.-J. Lee, Anchor text mining for translation of web queries: a transitive translation approach. ACM Trans. Inform. Syst. (TOIS) **22**, 242 (2004)
19. L.R. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition Proc. IEEE **77**, 257–286 (1989)
20. A.J. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Trans. Inform. Theory **13**, 260 (1967)