Learning Transliteration Lexicons from the Web

Jin-Shea Kuo[1, 2]
[1]Chung-Hwa Telecom. Laboratories, Taiwan
jskuo@cht.com.tw

Haizhou Li
Institute for Infocomm Research, Singapore
hzli@ieee.org

Ying-Kuei Yang[2]
[2]National Taiwan University of Science and Technology, Taiwan
ykyang@mouse.ee.ntust.edu.tw

Abstract

This paper presents an adaptive learning framework for Phonetic Similarity Modeling (PSM) that supports the automatic construction of transliteration lexicons. The learning algorithm starts with minimum prior knowledge about machine transliteration, and acquires knowledge iteratively from the Web. We study the active learning and the unsupervised learning strategies that minimize human supervision in terms of data labeling. The learning process refines the PSM and constructs a transliteration lexicon at the same time. We evaluate the proposed PSM and its learning algorithm through a series of systematic experiments, which show that the proposed framework is reliably effective on two independent databases.

1 Introduction

In applications such as cross-lingual information retrieval (CLIR) and machine translation (MT), there is an increasing need to translate out-of-vocabulary (OOV) words, for example from an alphabetical language to Chinese. Foreign proper names constitute a good portion of OOV words, which are translated into Chinese through transliteration. Transliteration is a process of translating a foreign word into a native language by preserving its pronunciation in the original language, otherwise known as *translation-by-sound*.

MT and CLIR systems rely heavily on bilingual lexicons, which are typically compiled manually. However, in view of the current information explosion, it is labor intensive, if not impossible, to compile a complete proper nouns lexicon. The Web is growing at a fast pace and is providing a live information source that is rich in transliterations. This paper presents a novel solution for automatically constructing an English-Chinese transliteration lexicon from the Web.

Research on automatic transliteration has reported promising results for *regular* transliteration (Wan and Verspoor, 1998; Li et al, 2004), where transliterations follow rigid guidelines. However, in Web publishing, translators in different countries and regions may not observe common guidelines. They often skew the transliterations in different ways to create special meanings to the sound equivalents, resulting in *casual* transliterations. In this case, the common generative models (Li et al, 2004) fail to predict the transliteration most of the time. For example, "Coca Cola" is transliterated into "可口可樂 /Ke-Kou-Ke-Le/" as a sound equivalent in Chinese, which literately means "happiness in the mouth". In this paper, we are interested in constructing lexicons that cover both *regular* and *casual* transliterations.

When a new English word is first introduced, many transliterations are invented. Most of them are *casual* transliterations because a regular transliteration typically does not have many variations. After a while, the transliterations converge into one or two popular ones. For example, "Taxi" becomes "的士 /Di-Shi/" in China and "德士 /De-Shi/" in Singapore. Therefore, the adequacy of a transliteration entry could be judged by its popularity and its conformity with the *translation-by-sound* principle. In any case, the phonetic similarity should serve as the primary basis of judgment.

This paper is organized as follows. In Section 2, we briefly introduce prior works pertaining to machine transliteration. In Section 3, we propose a phonetic similarity model (PSM) for confidence scoring of transliteration. In Section 4, we propose an adaptive learning process for PSM modeling and lexicon construction. In Section 5, we conduct experiments to evaluate different adaptive learning strategies. Finally, we conclude in Section 6.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 1129–1136, Sydney, July 2006.

2 Related Work

In general, studies of transliteration fall into two categories: transliteration modeling (TM) and extraction of transliteration pairs (EX) from corpora.

The TM approach models phoneme-based or grapheme-based mapping rules using a generative model that is trained from a large bilingual lexicon, with the objective of translating unknown words on the fly. The efforts are centered on establishing the phonetic relationship between transliteration pairs. Most of these works are devoted to phoneme[1]-based transliteration modeling (Wan and Verspoor 1998, Knight and Graehl, 1998). Suppose that *EW* is an English word and *CW* is its prospective Chinese transliteration. The phoneme-based approach first converts *EW* into an intermediate phonemic representation *P*, and then converts the phonemic representation into its Chinese counterpart *CW*. In this way, *EW* and *CW* form an *E-C* transliteration pair.

In this approach, we model the transliteration using two conditional probabilities, *P(CW|P)* and *P(P|EW)*, in a generative model *P(CW|EW) = P(CW|P)P(P|EW)*. Meng (2001) proposed a rule-based mapping approach. Virga and Khudanpur (2003) and Kuo et al (2005) adopted the noisy-channel modeling framework. Li et al (2004) took a different approach by introducing a joint source-channel model for direct orthography mapping (DOM), which treats transliteration as a statistical machine translation problem under monotonic constraints. The DOM approach, which is a grapheme-based approach, significantly outperforms the phoneme-based approaches in *regular* transliterations. It is noted that the state-of-the-art accuracy reported by Li et al (2004) for *regular* transliterations of the Xinhua database is about 70.1%, which leaves much room for improvement if one expects to use a generative model to construct a lexicon for *casual* transliterations.

EX research is motivated by information retrieval techniques, where people attempt to extract transliteration pairs from corpora. The EX approach aims to construct a large and up-to-date transliteration lexicon from live corpora. Towards this objective, some have proposed extracting translation pairs from parallel or comparable bitext using co-occurrence analysis or a context-vector approach (Fung and Yee, 1998; Nie et al, 1999). These methods compare the semantic similarities between words without taking their phonetic similarities into accounts. Lee and Chang (2003) proposed using a probabilistic model to identify *E-C* pairs from aligned sentences using phonetic clues. Lam et al (2004) proposed using semantic and phonetic clues to extract *E-C* pairs from comparable corpora. However, these approaches are subject to the availability of parallel or comparable bitext. A method that explores non-aligned text was proposed by harvesting *katakana*-English pairs from query logs (Brill et al, 2001). It was discovered that the unsupervised learning of such a transliteration model could be overwhelmed by noisy data, resulting in a decrease in model accuracy.

Many efforts have been made in using Web-based resources for harvesting transliteration/ translation pairs. These include exploring query logs (Brill et al, 2001), unrelated corpus (Rapp, 1999), and parallel or comparable corpus (Fung and Yee, 1998; Nie et al, 1999; Huang et al 2005). To establish correspondence, these algorithms usually rely on one or more statistical clues, such as the correlation between word frequencies, cognates of similar spelling or pronunciations. They include two aspects. First, a robust mechanism that establishes statistical relationships between bilingual words, such as a phonetic similarity model which is motivated by the TM research; and second, an effective learning framework that is able to adaptively discover new events from the Web. In the prior work, most of the phonetic similarity models were trained on a static lexicon. In this paper, we address the EX problem by exploiting a novel Web-based resource. We also propose a phonetic similarity model that generates confidence scores for the validation of *E-C* pairs.

In Chinese webpages, translated or transliterated terms are frequently accompanied by their original Latin words. The latter serve as the appositives of the former. A sample search result for the query submission "Kuro" is the bilingual snippet[2] "...經營 Kuro 庫洛 P2P 音樂交換軟體的飛行網，3 日發表 P2P 與版權爭議的解決方案— C2C (Content to Community)...". The co-occurrence statistics in such a snippet was shown to be useful in constructing a transitive translation model (Lu et al, 2002). In the

[1] Both phoneme and syllable based approaches are referred to as phoneme-based here.

[2] A bilingual snippet refers to a Chinese predominant text with embedded English appositives.

example above, "Content to Community" is not a transliteration of C2C, but rather an acronym expansion, while "庫洛 /Ku-Luo/", as underlined, presents a transliteration for "Kuro". What is important is that the *E-C* pairs are always closely collocated. Inspired by this observation, we propose an algorithm that searches over the close context of an English word in a bilingual snippet for the word's transliteration candidates.

The contributions of this paper include: (i) an approach to harvesting real life *E-C* transliteration pairs from the Web; (ii) a phonetic similarity model that evaluates the confidence of so extracted *E-C* pair candidates; (iii) a comparative study of several machine learning strategies.

3 Phonetic Similarity Model

English and Chinese have different syllable structures. Chinese is a syllabic language where each Chinese character is a syllable in either consonant-vowel (CV) or consonant-vowel-nasal (CVN) structure. A Chinese word consists of a sequence of characters, phonetically a sequence of syllables. Thus, in first *E-C* transliteration, it is a natural choice to syllabify an English word by converting its phoneme sequence into a sequence of Chinese-like syllables, and then convert it into a sequence of Chinese characters.

There have been several effective algorithms for the syllabification of English words for transliteration. Typical syllabification algorithms first convert English graphemes to phonemes, referred to as the letter-to-sound transformation, then syllabify the phoneme sequence into a syllable sequence. For this method, a letter-to-sound conversion is needed (Pagel, 1998; Jurafsky, 2000). The phoneme-based syllabification algorithm is referred to as PSA. Another syllabification technique attempts to map the grapheme of an English word to syllables directly (Kuo and Yang, 2004). The grapheme-based syllabification algorithm is referred to as GSA. In general, the size of a phoneme inventory is smaller than that of a grapheme inventory. The PSA therefore requires less training data for statistical modeling (Knight, 1998); on the other hand, the grapheme-based method gets rid of the letter-to-sound conversion, which is one of the main causes of transliteration errors (Li et al, 2004).

Assuming that Chinese transliterations always co-occur in proximity to their original English words, we propose a phonetic similarity modeling (PSM) that measures the phonetic similarity between candidate transliteration pairs. In a bilingual snippet, when an English word *EW* is spotted, the method searches for the word's possible Chinese transliteration *CW* in its neighborhood. *EW* can be a single word or a phrase of multiple English words. Next, we formulate the PSM and the estimation of its parameters.

3.1 Generative Model

Let $ES = \{es_1, ...es_m, ...es_M\}$ be a sequence of English syllables derived from *EW*, using the PSA or GSA approach, and $CS = \{cs_1, ...cs_n, ...cs_N\}$ be the sequence of Chinese syllables derived from *CW*, represented by a Chinese character string $CW \rightarrow c_1, ...c_n, ..., c_N$. *EW* and *CW* is a transliteration pair. The *E-C* transliteration can be considered a generative process formulated by the noisy channel model, with *EW* as the input and *CW* as the output. $P(EW/CW)$ is estimated to characterize the noisy channel, known as the transliteration probability. $P(CW)$ is a language model to characterize the source language. Applying Bayes' rule, we have

$$P(CW/EW) = P(EW/CW)P(CW)/P(EW) \quad (1)$$

Following the *translation-by-sound* principle, the transliteration probability $P(EW/CW)$ can be approximated by the phonetic confusion probability $P(ES/CS)$, which is given as

$$P(ES/CS) = \max_{\Delta \in \Phi} P(ES, \Delta / CS), \quad (2)$$

where Φ is the set of all possible alignment paths between *ES* and *CS*. It is not trivial to find the best alignment path Δ. One can resort to a dynamic programming algorithm. Assuming conditional independence of syllables in *ES* and *CS*, we have $P(ES/CS) = \prod_{m=1}^{M} p(es_m / cs_m)$ in a special case where $M = N$. Note that, typically, we have $N \le M$ due to syllable elision. We introduce a null syllable φ and a dynamic warping strategy to evaluate $P(ES/CS)$ when $M \ne N$ (Kuo et al, 2005). With the phonetic approximation, Eq.(1) can be rewritten as

$$P(CW/EW) \approx P(ES/CS)P(CW)/P(EW) \quad (3)$$

The language model in Eq.(3) can be represented by Chinese characters *n*-gram statistics.

$$P(CW) = \prod_{n=1}^{N} p(c_n / c_{n-1}, c_{n-2}, ..., c_1) \quad (4)$$

In adopting bigram, Eq.(4) is rewritten as $P(CW) \approx p(c_1)\prod_{n=2}^{N} p(c_n / c_{n-1})$. Note that the context of *EW* usually has a number of competing Chinese transliteration candidates in a set, denoted as Ω . We rank the candidates by Eq.(1) to find the most likely *CW* for a given *EW*. In this process, $P(EW)$ can be ignored because it is the same for all *CW* candidates. The *CW* candidate that gives the highest posterior probability is considered the most probable candidate CW' .

$$\begin{aligned} CW' &= \arg\max_{CW \in \Omega} P(CW / EW) \\ &\approx \arg\max_{CW \in \Omega} P(ES / CS) P(CW) \end{aligned} \quad (5)$$

However, the most probable CW' isn't necessarily the desired transliteration. The next step is to examine if CW' and *EW* indeed form a genuine *E-C* pair. We define the confidence of the *E-C* pair as the posterior odds similar to that in a hypothesis test under the Bayesian interpretation. We have H_0 , which hypothesizes that CW' and *EW* form an *E-C* pair, and H_1 , which hypothesizes otherwise. The posterior odds is given as follows,

$$\sigma = \frac{P(H_0 / EW)}{P(H_1 / EW)} \approx \frac{P(ES / CS')P(CW')}{\sum_{\substack{CW \in \Omega \\ CW \neq CW'}} P(ES / CS)P(CW)} \quad (6)$$

where CS' is the syllable sequence of CW' , $p(H_1 / EW)$ is approximated by the probability mass of the competing candidates of CW' , or $\sum_{\substack{CW \in \Omega \\ CW \neq CW'}} P(ES / CS)P(CW)$. The higher the σ is, the more probable that hypothesis H_0 overtakes H_1 . The PSM formulation can be seen as an extension to prior work (Brill et al, 2001) in transliteration modeling. We introduce the posterior odds σ as the confidence score so that *E-C* pairs that are extracted from different contexts can be directly compared. In practice, we set a threshold for σ to decide a cutoff point for *E-C* pairs short-listing.

3.2 PSM Estimation

The PSM parameters are estimated from the statistics of a given transliteration lexicon, which is a collection of manually selected *E-C* pairs in supervised learning, or a collection of high confidence *E-C* pairs in unsupervised learning. An initial PSM is bootstrapped using prior knowledge such as rule-based syllable mapping. Then we align the *E-C* pairs with the PSM and derive syllable mapping statistics for PSA and GSA syllabifications. A final PSM is a linear combination of the PSA-based PSM (PSA-PSM) and the GSA-based PSM (GSA-PSM). The PSM parameter $p(es_m / cs_n)$ can be estimated by an *Expectation-Maximization* (EM) process (Dempster, 1977). In the *Expectation* step, we compute the counts of events such as $\#<es_m, cs_n>$ and $\#<cs_n>$ by force-aligning the *E-C* pairs in the training lexicon Ψ . In the *Maximization* step, we estimate the PSM parameters $p(es_m / cs_n)$ by

$$p(es_m / cs_n) = \#<es_m, cs_n> / \#<cs_n> . \quad (7)$$

As the EM process guarantees non-decreasing likelihood probability $\prod_{\forall \Psi} P(ES / CS)$, we let the EM process iterate until $\prod_{\forall \Psi} P(ES / CS)$ converges. The EM process can be thought of as a refining process to obtain the best alignment between the *E-C* syllables and at the same time a re-estimating process for PSM parameters. It is summarized as follows.

Start: Bootstrap PSM parameters $p(es_m / cs_n)$ using prior phonetic mapping knowledge

E-Step: Force-align corpus Ψ using existing $p(es_m / cs_n)$ and compute the counts of $\#<es_m, cs_n>$ and $\#<cs_n>$;

M-Step: Re-estimate $p(es_m / cs_n)$ using the counts from E-Step.

Iterate: Repeat E-Step and M-Step until $\prod_{\forall \Psi} P(ES / CS)$ converges.

4 Adaptive Learning Framework

We propose an adaptive learning framework under which we learn PSM and harvest *E-C* pairs from the Web at the same time. Conceptually, the adaptive learning is carried out as follows.

We obtain bilingual snippets from the Web by iteratively submitting queries to the Web search engines (Brin and Page, 1998). For each batch of querying, the query results are all normalized to plain text, from which we further extract qualified sentences. A qualified sentence has at least one English word. Under this criterion, a collection of qualified sentences can be extracted automatically. To label the *E-C* pairs, each qualified sentence is manually checked based on the following transliteration criteria: (i) if an *EW* is partly translated phonetically and partly translated semantically, only the phonetic transliteration constituent is extracted to form a

transliteration pair; (ii) elision of English sound is accepted; (iii) multiple *E-C* pairs can appear in one sentence; (iv) an *EW* can have multiple valid Chinese transliterations and vice versa. The validation process results in a collection of qualified *E-C* pairs, also referred to as Distinct Qualified Transliteration Pairs (DQTPs).

As formulated in Section 3, the PSM is trained using a training lexicon in a data driven manner. It is therefore very important to ensure that in the learning process we have prepared a quality training lexicon. We establish a baseline system using supervised learning. In this approach, we use human labeled data to train a model. The advantage is that it is able to establish a model quickly as long as labeled data are available. However, this method also suffers from some practical issues. First, the derived model can only be as good as the data that it sees. An adaptive mechanism is therefore needed for the model to acquire new knowledge from the dynamically growing Web. Second, a massive annotation of database is labor intensive, if not entirely impossible.

To reduce the annotation needed, we discuss three adaptive strategies cast in the machine learning framework, namely active learning, unsupervised learning and active-unsupervised learning. The learning strategies can be depicted in Figure 1 with their difference being discussed next. We also train a baseline system using supervised learning approach as a reference point for benchmarking purpose.

4.1 Active Learning

Active learning is based on the assumption that a small number of labeled samples, which are DQTPs here, and a large number of unlabeled samples are available. This assumption is valid in most NLP tasks. In contrast to supervised learning, where the entire corpus is labeled manually, active learning selects the most useful samples for labeling and adds the labeled examples to the training set to retrain the model. This procedure is repeated until the model achieves a certain level of performance. Practically, a batch of samples is selected each time. This is called batch-based sample selection (Lewis and Catlett, 1994), as shown in the search and ranking block in Figure 1.

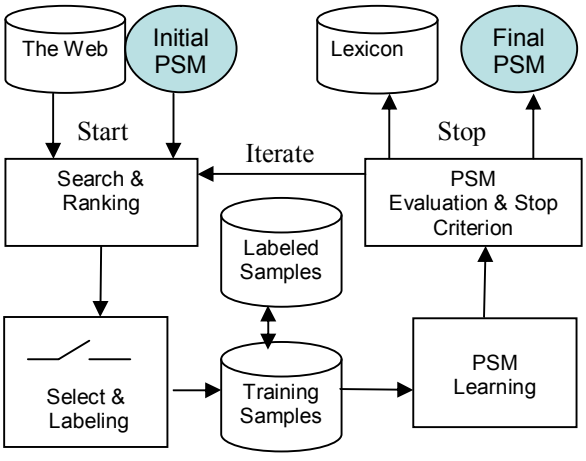

Figure 1. An adaptive learning framework for automatic construction of transliteration lexicon.

For an active learning to be effective, we propose using three measures to select candidates for human labeling. First, we would like to select the most uncertain samples that are potentially highly informative for the PSM model. The informativeness of a sample can be quantified by its confidence score σ as in the PSM formulation. Ranking the *E-C* pairs by σ is referred to as C-rank. The samples of low C-rank are the interesting samples to be labeled. Second, we would like to select candidates that are of low frequency. Ranking by frequency is called F-rank. During Web crawling, most of the search engines use various strategies to prevent spamming and one of fundamental tasks is to remove the duplicated Web pages. Therefore, we assume that the bilingual snippets are all unique. Intuitively, *E-C* pairs of low frequency indicate uncommon events which are of higher interest to the model. Third, we would like to select samples upon which the PSA-PSM and GSA-PSM disagree the most. The disagreed upon samples represent new knowledge to the PSM. In short, we select low C-rank, low F-rank and PSM-disagreed samples for labeling because the high C-rank, high F-rank and PSM-agreed samples are already well known to the model.

4.2 Unsupervised Learning

Unsupervised learning skips the human labeling step. It minimizes human supervision by automatically labeling the data. This can be effective if prior knowledge about a task is available, for example, if an initial PSM can be built based on human crafted phonetic mapping rules. This is entirely possible. Kuo et al (2005) proposed using a cross-lingual phonetic confusion matrix resulting from automatic speech recognition to bootstrap an initial PSM model. The task of labeling samples is basically to distinguish the qualified transliteration pairs from the rest. Unlike the sample selection method in active learning, here we would like to

select the samples that are of high C-rank and high F-rank because they are more likely to be the desired transliteration pairs.

The difference between the active learning and the unsupervised learning strategies lies in that the former selects samples for human labeling, such as in the select & labeling block in Figure 1 before passing on for PSM learning, while the latter selects the samples automatically and assumes they are all correct DQTPs. The disadvantage of unsupervised learning is that it tends to reinforce its existing knowledge rather than to discover new events.

4.3 Active-Unsupervised Learning

The active learning and the unsupervised learning strategies can be complementary. Active learning minimizes the labeling effort by intelligently short-listing informative and representative samples for labeling. It makes sure that the PSM learns new and informative knowledge over iterations. Unsupervised learning effectively exploits the unlabelled data. It reinforces the knowledge that PSM has acquired and allows PSM to adapt to changes at no cost. However, we do not expect unsupervised learning to acquire new knowledge like active learning does. Intuitively, a better solution is to integrate the two strategies into one, referred to as the active-unsupervised learning strategy. In this strategy, we use active learning to select a small amount of informative and representative samples for labeling. At the same time, we select samples of high confidence score from the rest and consider them correct *E-C* pairs. We then merge the labeled set with the high-confidence set in the PSM re-training.

5 Experiments

We first construct a development corpus by crawling of webpages. This corpus consists of about 500 MB of webpages, called SET1 (Kuo et al, 2005). Out of 80,094 qualified sentences, 8,898 DQTPs are manually extracted from SET1, which serve as the gold standard in testing. To establish a baseline system, we first train a PSM using all 8,898 DQTPs in supervised manner and conduct a closed test on SET1 as in Table 1. We further implement three PSM learning strategies and conduct a systematic series of experiments.

	Precision	Recall	F-measure
closed-test	0.79	0.69	0.74

Table 1. Supervised learning test on SET1

5.1 Unsupervised Learning

We follow the formulation described in Section 4.2. First, we derive an initial PSM using randomly selected 100 seed DQTPs and simulate the Web-based learning process with the SET1: (i) select high F-rank and high C-rank *E-C* pairs using PSM, (ii) add the selected *E-C* pairs to the DQTP pool as if they are true DQTPs, and (iii) reestimate PSM by using the updated DQTP pool.

In Figure 2, we report the F-measure over iterations. The U_HF curve reflects the learning progress of using *E-C* pairs that occur more than once in the SET1 corpus (high F-rank). The U_HF_HR curve reflects the learning progress using a subset of *E-C* pairs from U_HF which has high posterior odds as defined in Eq.(6). Both selection strategies aim to select *E-C* pairs, which are as genuine as possible.

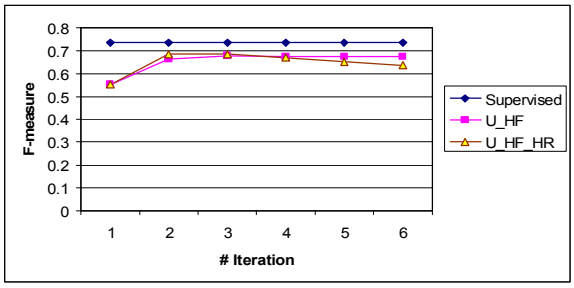

Figure 2. F-measure over iterations for unsupervised learning on SET1.

We found that both U_HF and U_HF_HR give similar results in terms of F-measure. Without surprise, more iterations don't always lead to better performance because unsupervised learning doesn't aim to acquiring new knowledge over iterations. Nevertheless, unsupervised learning improves the initial PSM in the first iteration substantially. It can serve as an effective PSM adaptation method.

5.2 Active Learning

The objective of active learning is to minimize human supervision by automatically selecting the most informative samples to be labeled. The effect of active learning is that it maximizes performance improvement with minimum annotation effort. Like in unsupervised learning, we start with the same 100 seed DQTPs and an initial PSM model and carry out experiments on SET1: (i) select low F-rank, low C-rank and GSA-PSM and PSA-PSM disagreed *E-C* pairs; (ii) label the selected pairs by removing the non-*E-C* pairs and add the labeled *E-C* pairs to the DQTP pool, and (iii) reestimate the PSM by using the updated DQTP pool.

To select the samples, we employ 3 different strategies: A_LF_LR, where we only select low F-rank and low C-rank candidates for labeling. A_DIFF, where we only select those that GSA-PSM and PSA-PSM disagreed upon; and A_DIFF_LF_LR, the union of A_LF_LR and A_DIFF selections. As shown in Figure 3, the F-measure of A_DIFF (0.729) and A_DIFF_LF_LR (0.731) approximate to that of supervised learning 0.735) after four iterations.

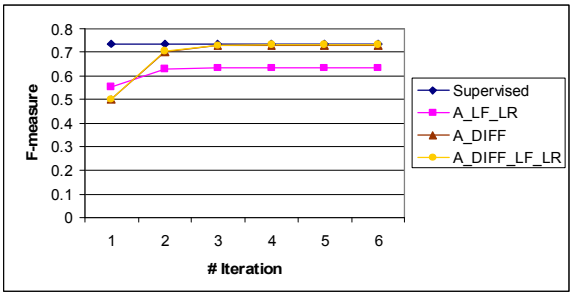

Figure 3. F-measure over iterations for active learning on SET1.

With almost identical performance as supervised learning, the active learning approach has greatly reduced the number of samples for manual labeling as reported in Table 2. It is found that for active learning to reach the performance of supervised learning, A_DIFF is the most effective strategy. It reduces the labeling effort by 89.0%, from 80,094 samples to 8,750.

	Sample selection	#samples labeled
Active learning	A_LF_LR	1,671
	A_DIFF	8,750
	A_DIFF_LF_LR	9,683
Supervised learning		80,094

Table 2. Number of total samples for manual labeling in 6 iterations of Figure 3.

5.3 Active Unsupervised Learning

It would be interesting to study the performance of combining unsupervised learning and active learning. The experiment is similar to that of active learning except that, in step (iii) of active learning, we take the unlabeled *high confidence* candidates (high F-rank and high C-rank as in U_HF_HR of Section 5.1) as the true labeled samples and add into the DQTP pool. The result is shown in Figure 4. Although active unsupervised learning was reported having promising results (Riccardi and Hakkani-Tur, 2003) in some NLP tasks, it has not been as effective as active learning alone in this experiment probably due to the fact the unlabeled *high confidence* candidates are still too noisy to be informative.

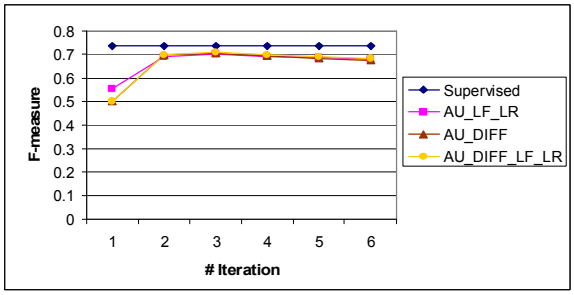

Figure 4. F-measure over iterations for active unsupervised learning on SET1.

5.4 Learning Transliteration Lexicons

The ultimate objective of building a PSM is to extract a transliteration lexicon from the Web by iteratively submitting queries and harvesting new transliteration pairs from the return results until no more new pairs. For example, by submitting "Robert" to search engines, we may get "Robert-羅伯特", "Richard-理查" and "Charles-查爾斯" in return. In this way, new queries can be generated iteratively, thus new pairs are discovered. We pick the best performing SET1-derived PSM trained using A_DIFF_LF_LR active learning strategy and test it on a new database SET2 which is obtained in the same way as SET1.

	Before adaptation	After adaptation
#distinct *E-C* pairs	137,711	130,456
Precision	0.777	0.846
#expected DQTPs	107,001	110,365

Table 3. SET1-derived PSM adapted towards SET2.

SET2 contains 67,944 Web pages amounting to 3.17 GB. We extracted 2,122,026 qualified sentences from SET2. Using the PSM, we extract 137,711 distinct *E-C* pairs. As the gold standard for SET2 is unavailable, we randomly select 1,000 pairs for manual checking. A precision of 0.777 is reported. In this way, 107,001 DQTPs can be expected. We further carry out one iteration of unsupervised learning using U_HF_HR to adapt the SET1-derived PSM towards SET2. The results before and after adaptation are reported in Table 3. Like the experiment in Section 5.1, the unsupervised learning improves the PSM in terms of precision significantly.

6 Conclusions

We have proposed a framework for harvesting *E-C* transliteration lexicons from the Web using bilingual snippets. In this framework, we formulate the PSM learning and *E-C* pair evaluation methods. We have studied three strategies for PSM learning aiming at reducing the human supervision.

The experiments show that unsupervised learning is an effective way for rapid PSM adaptation while active learning is the most effective in achieving high performance. We find that the Web is a resourceful live corpus for real life *E-C* transliteration lexicon learning, especially for casual transliterations. In this paper, we use two Web databases SET1 and SET2 for simplicity. The proposed framework can be easily extended to an incremental learning framework for live databases. This paper has focused solely on use of phonetic clues for lexicon and PSM learning. We have good reason to expect the combining semantic and phonetic clues to improve the performance further.

References

E. Brill, G. Kacmarcik, C. Brockett. 2001. Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs, In *Proc. of NLPPRS*, pp. 393-399.

S. Brin and L. Page. 1998. The Anatomy of a Large-scale Hypertextual Web Search Engine, In *Proc. of 7th WWW,* pp. 107-117.

A. P. Dempster, N. M. Laird and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm, Journal of the Royal Statistical Society, Ser. B. Vol. 39, pp. 1-38.

P. Fung and L.-Y. Yee. 1998. An IR Approach for Translating New Words from Nonparallel, Comparable Texts. In *Proc. of 17th COLING and 36th ACL,* pp. 414-420.

F. Huang, Y. Zhang and Stephan Vogel. 2005. Mining Key Phrase Translations from Web Corpora. In *Proc. of HLT-EMNLP*, pp. 483-490.

D. Jurafsky and J. H. Martin. 2000. Speech and Language Processing, pp. 102-120, Prentice-Hall, New Jersey.

K. Knight and J. Graehl. 1998. Machine Transliteration, Computational Linguistics, Vol. 24, No. 4, pp. 599-612.

J.-S. Kuo and Y.-K. Yang. 2004. Constructing Transliterations Lexicons from Web Corpora, In *the Companion Volume, 42nd ACL*, pp. 102-105.

J.-S. Kuo and Y.-K. Yang. 2005. Incorporating Pronunciation Variation into Extraction of Transliterated-term Pairs from Web Corpora, In *Proc. of ICCC*, pp. 131-138.

C.-J. Lee and J.-S. Chang. 2003. Acquisition of English-Chinese Transliterated Word Pairs from Parallel-Aligned Texts Using a Statistical Machine Transliteration Model, In *Proc. of HLT-NAACL Workshop Data Driven MT and Beyond*, pp. 96-103.

D. D. Lewis and J. Catlett. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning, In *Proc. of ICML 1994*, pp. 148-156.

H. Li, M. Zhang and J. Su. 2004. A Joint Source Channel Model for Machine Transliteration, In *Proc. of 42nd ACL*, pp. 159-166.

W. Lam, R.-Z. Huang and P.-S. Cheung. 2004. Learning Phonetic Similarity for Matching Named Entity Translations and Mining New Translations, In *Proc. of 27th ACM SIGIR*, pp. 289-296.

W.-H. Lu, L.-F. Chien and H.-J Lee. 2002. Translation of Web Queries Using Anchor Text Mining, TALIP, Vol. 1, Issue 2, pp. 159- 172.

H. M. Meng, W.-K. Lo, B. Chen and T. Tang. 2001. Generate Phonetic Cognates to Handle Name Entities in English-Chinese Cross-Language Spoken Document Retrieval, In *Proc. of ASRU*, pp. 311-314.

J.-Y. Nie, P. Isabelle, M. Simard, and R. Durand. 1999. Cross-language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Text from the Web", In *Proc. of 22nd ACM SIGIR*, pp 74-81.

V. Pagel, K. Lenzo and A. Black. 1998. Letter to Sound Rules for Accented Lexicon Compression, In *Proc. of ICSLP*, pp. 2015-2020.

R. Rapp. 1999. Automatic Identification of Word Translations from Unrelated English and German Corpora, In *Proc. of 37th ACL*, pp. 519-526.

G. Riccardi and D. Hakkani-Tür. 2003. Active and Unsupervised Learning for Automatic Speech Recognition. In *Proc. of 8th Eurospeech*.

P. Virga and S. Khudanpur. 2003. Transliteration of Proper Names in Cross-Lingual Information Retrieval, In *Proc. of 41st ACL Workshop on Multilingual and Mixed Language Named Entity Recognition*, pp. 57-64.

S. Wan and C. M. Verspoor. 1998. Automatic English-Chinese Name Transliteration for Development of Multilingual Resources, In *Proc. of 17th COLING and 36th ACL*, pp.1352-1356.