Multi-View Co-training of Transliteration Model

Jin-Shea Kuo
Chung-Hwa Telecomm.
Laboratories, Taiwan
d8807302@gmail.com

Haizhou Li
Institute for Infocomm Research,
Singapore 119613
hli@i2r.a-star.edu.sg

Abstract

This paper discusses a new approach to training of transliteration model from unlabeled data for transliteration extraction. We start with an inquiry into the formulation of transliteration model by considering different transliteration strategies as a multi-view problem, where each view exploits a natural division of transliteration features, such as phoneme-based, grapheme-based or hybrid features. Then we introduce a multi-view Co-training algorithm, which leverages *compatible* and partially *uncorrelated* information across different views to effectively boost the model from unlabeled data. Applying this algorithm to transliteration extraction, the results show that it not only circumvents the need of data labeling, but also achieves performance close to that of supervised learning, where manual labeling is required for all training samples.

1 Introduction

Named entities are important content words in text documents. In many applications, such as cross-language information retrieval (Meng et al., 2001; Virga and Khudanpur, 2003) and machine translation (Knight and Graehl, 1998; Chen et al., 2006), one of the fundamental tasks is to identify these words. Imported foreign proper names constitute a good portion of such words, which are newly translated into Chinese by transliteration. Transliteration is a process of translating a foreign word into the native language by preserving its pronunciation in the original language, otherwise known as *translation-by-sound*.

As new words emerge everyday, no lexicon is able to cover all transliterations. It is desirable to find ways to harvest transliterations from real world corpora. In this paper, we are interested in the learning of English to Chinese (*E-C*) transliteration model for transliteration extraction from the Web.

A statistical transliteration model is typically trained on a large amount of transliteration pairs, also referred to a bilingual corpus. The correspondence between a transliteration pair may be described by the mapping of different basic pronunciation units (BPUs) such as phoneme-based[1], or grapheme-based one, or both. We can see each type of BPU mapping as a natural division of transliteration features, which represents a view to the phonetic mapping problem. By using different BPUs, we approach the transliteration modeling and extraction problems from different views.

This paper is organized as follows. In Section 2, we briefly introduce previous work. In Section 3, we conduct an inquiry into the formulation of transliteration model or phonetic similarity model (PSM) and consider it as a multi-view problem. In Section 4, we propose a multi-view Co-training strategy for PSM training and transliteration extraction. In Section 5, we study the effectiveness of proposed algorithms. Finally, we conclude in Section 6.

2 Related Work

Studies on transliteration have been focused on transliteration modeling and transliteration extraction. The transliteration modeling approach deduces either phoneme-based or grapheme-based mapping rules using a generative model that is

[1] Both phoneme and syllable based approaches are referred to as phoneme-based in this paper.

trained from a large bilingual corpus. Most of the works are devoted to phoneme-based transliteration modeling (Knight and Graehl, 1998; Lee, 1999). Suppose that *EW* is an English word and *CW* is its Chinese transliteration. *EW* and *CW* form an *E-C* transliteration pair. The phoneme-based approach first converts *EW* into an intermediate phonemic representation *p*, and then converts *p* into its Chinese counterpart *CW*. The idea is to transform both source and target words into comparable phonemes so that the phonetic similarity between two words can be measured easily.

Recently the grapheme-based approach has attracted much attention. It was proposed by Jeong et al. (1999), Li et al. (2004) and many others (Oh et al., 2006b), which is also known as direct orthography mapping. It treats the transliteration as a statistical machine translation problem under monotonic constraint. The idea is to obtain the bilingual orthographical correspondence directly to reduce the possible errors introduced in multiple conversions. However, the grapheme-based transliteration model has more parameters than phoneme-based one does, thus expects a larger training corpus.

Most of the reported works have been focused on either phoneme- or grapheme-based approaches. Bilac and Tanaka (2004) and Oh et al. (2006a; 2006b) recently proposed using a mix of phoneme and grapheme features, where both features are fused into a single learning process. The feature fusion was shown to be effective. However, their methods hinge on the availability of a labeled bilingual corpus.

In transliteration extraction, mining translations or transliterations from the ever-growing multilingual Web has become an active research topic, for example, by exploring query logs (Brill et al., 2001) and parallel (Nie et al., 1999) or comparable corpora (Sproat et al., 2006). Transliterations in such a live corpus are typically unlabeled. For model-based transliteration extraction, recent progress in machine learning offers different options to exploit unlabeled data, that include active learning (Lewis and Catlett, 1994) and Co-training (Nigam and Ghani, 2000; Tür et al. 2005).

Taking the prior work a step forward, this paper explores a new way of fusing phoneme and grapheme features through a multi-view Co-training algorithm (Blum and Mitchell, 1998), which starts with a small number of labeled data to bootstrap a transliteration model to automatically harvest transliterations from the Web.

3 Phonetic Similarity Model with Multiple Views

Machine transliteration can be formulated as a generative process, which takes a character string in source language as input and generates a character string in the target language as output. Conceptually, this process can be regarded as a 3-step decoding: segmentation of both source and target strings into basic pronunciation units (BPUs), relating the source BPUs with target units by resolving different combinations of alignments and unit mappings in finding the most probable BPU pairs. A BPU can be defined as a phoneme sequence, a grapheme sequence, or a part of them. A transliteration model establishes the phonetic relationship between BPUs in two languages to measure their similarity, therefore, it is also known as the phonetic similarity model (PSM).

To introduce the multi-view concept, we illustrate the BPU transfers in Figure 1, where each transfer is represented by a direct path with different line style. There are altogether four different paths: the phoneme-based path V_1 ($T_1 \rightarrow T_2 \rightarrow T_3$), the grapheme-based path V_4 (T_4), and their variants, V_2($T_1 \rightarrow T_5$) and V_3($T_6 \rightarrow T_3$). The last two paths make use of the intermediate BPU mappings between phonemes and graphemes. Each of the paths represents a view to the mapping problem. Given a labeled bilingual corpus, we are able to train a transliteration model for each view easily.

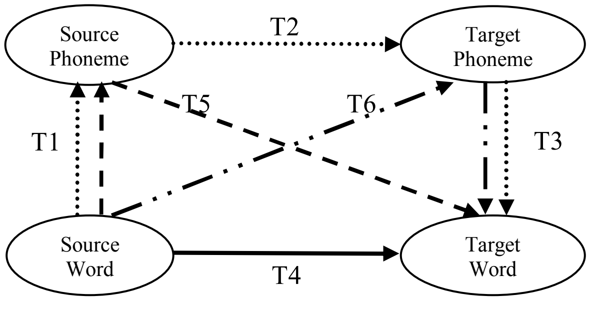

Figure 1. Multiple views for establishing transliteration correspondence.

The *E-C* transliteration has been studied extensively in the paradigm of noisy channel model

(Manning and Scheutze, 1999), with *EW* as the observation and *CW* as the input to be recovered. Applying Bayes rule, the transliteration can be described by Eq. (1),

$$P(CW \mid EW) = \frac{P(EW \mid CW) \times P(CW)}{P(EW)}, \qquad (1)$$

where we need to deal with two probability distributions: *P(EW|CW)*, the probability of transliterating *CW* to *EW*, also known as the unit mapping rules, and *P(CW)*, the probability distribution of *CW*, known as the target language model.

Representing *EW* in English BPU sequence $EP = \{ep_1, ...ep_m, ...ep_M\}$ and *CW* in Chinese one, $CP = \{cp_1, ...cp_n, ...cp_N\}$, a typical transliteration probability can be expressed as,

$$P(EW \mid CW) \approx P(EW \mid EP) \times P(EP \mid CP) \times P(CP \mid CW). \quad (2)$$

The language model, *P(CW)*, can be represented by Chinese characters *n*-gram statistics (Manning and Scheutze, 1999) and expressed in Eq. (3). In the case of bigram, we have,

$$P(CW) \approx P(c_1) \prod_{n=2}^{N} P(c_n \mid c_{n-1}) \qquad (3)$$

We next rewrite Eq. (2) for the four different views depicted in Figure 1 in a systematic manner.

3.1 Phoneme-based Approach

The phoneme-based approach approximates the transliteration probability distribution by introducing an intermediate phonemic representation. In this way, we convert the words in the source language, say $EW = e_1, e_2 ... e_K$, into English syllables ES , then Chinese syllables CS and finally the target language, say Chinese $CW = c_1, c_2 ... c_K$ in sequence. Eq. (2) can be rewritten by replacing *EP* and *CP* with *ES* and *CS*, respectively, and expressed by Eq. (4).

$$P(EW \mid CW) \approx P(EW \mid ES) \times P(ES \mid CS) \times P(CS \mid CW) \qquad (4)$$

The three probabilities correspond to the three-step mapping in V_1 path.

The phoneme-based approach suffers from multiple step mappings. This could compromise overall performance because none of the three steps guarantees a perfect conversion.

3.2 Grapheme-based Approach

The grapheme-based approach is inspired by the transfer model (Vauqois, 1988) in machine translation that estimates $P(EW \mid CW)$ directly without interlingua representation. This method aims to alleviate the imprecision introduced by the multiple transfers in phoneme-based approach.

In practice, a grapheme-based approach converts the English graphemes to Chinese graphemes in one single step. Suppose that we have $EW = e_1, e_2 ... e_K$ and $CW = c_1, c_2 ... c_K$ where e_k and c_k are aligned grapheme units.

Under the noisy channel model, we can estimate $P(EW \mid CW)$ based on the alignment statistics which is similar to the lexical mapping in statistical machine translation.

$$P(EW \mid CW) \approx \prod_{k=1}^{K} P(e_k \mid c_k) \qquad (5)$$

Eq.(5) is a grapheme-based alternative to Eq.(2).

3.3 Hybrid Approach

A tradeoff between the phoneme- and grapheme-based approaches is to take shortcuts to the mapping between phonemes and graphemes of two languages via V_2 or V_3, where only two steps of mapping are involved. For V_3, we rewrite Eq.(2) as Eq. (6):

$$P(EW \mid CW) = P(EW \mid CS) \times P(CS \mid CW), \qquad (6)$$

where $P(EW \mid CS)$ translates Chinese sounds into English words. For V_2, we rewrite Eq. (2) as Eq. (7):

$$P(EW \mid CW) = P(EW \mid ES) \times P(ES \mid CW), \qquad (7)$$

where $P(ES \mid CW)$ translates Chinese words into English sounds.

Eqs. (4) – (7) describe the four paths of transliteration. In a multi-view problem, one partitions the domain's features into subsets, each of which is sufficient for learning the target concept. Here the target concept is the label of transliteration pair. Given a collection of *E-C* pair candidates, the transliteration extraction task can be formulated as a hypothesis test, which makes a binary decision as to whether a candidate *E-C* pair is a genuine transliteration pair or not. Given an *E-C* pair *X*={*EW*,*CW*}, we have H_0 , which hypothesizes that EW and CW form a genuine *E-C* pair, and H_1, which hypothesizes otherwise. The likelihood ratio is given as $\sigma = P(X \mid H_0) / P(X \mid H_1)$, where $P(X \mid H_0)$ and $P(X \mid H_0)$ are derived from *P(EW|CW)*. By comparing σ with a threshold τ, we make the binary decision as that in (Kuo et al., 2007).

As discussed, each view takes a distinct path that has its own advantages and disadvantages in terms of model expressiveness and complexity. Each view represents a weak learner achieving moderately good performance towards the target concept. Next, we study a multi-view Co-training process that leverages the data of different views from each other in order to boost the accuracy of a PSM model.

4 Multi-View Learning Framework

The PSM can be trained in a supervised manner using a manually labeled corpus. The advantage of supervised learning is that we can establish a model quickly as long as labeled data are available. However, this method suffers from some practical constraints. First, the derived model can only be as good as the data it sees. Second, the labeling of corpus is labor intensive.

To circumvent the need of manual labeling, here we study three adaptive strategies cast in the machine learning framework, namely unsupervised learning, Co-training and Co-EM.

4.1 Unsupervised Learning

Unsupervised learning minimizes human supervision by probabilistically labeling data through an Expectation and Maximization (EM) (Dempster et al., 1977) process. The unsupervised learning strategy can be depicted in Figure 2 by taking the dotted path, where the extraction process accumulates all the acquired transliteration pairs in a repository for training a new PSM. A new PSM is in turn used to extract new transliteration pairs. The unsupervised learning approach only needs a few labeled samples to bootstrap the initial model for further extraction. Note that the training samples are noisy and hence the quality of initial PSM therefore has a direct impact on the final performance.

4.2 Co-training and Co-EM

The multi-view setting (Muslea et al., 2002) applies to learning problems that have a natural way to divide their features into different views, each of which is sufficient to learn the target concept. Blum and Mitchell (1998) proved that for a problem with two views, the target concept can be learned based on a few labeled and many unlabeled examples, provided that the views are *compatible* and *uncorrelated*. Intuitively, the transliteration problem has *compatible* views. If an *E-C* pair forms a transliteration, then this is true across all different views. However, it is arguable that the four views in Figure 1 are *uncorrelated*. Studies (Nigam and Ghani, 2000; Muslea et al., 2002) shown that the views do not have to be entirely *uncorrelated* for Co-training to take effect. This motivates our attempt to explore multi-view Co-training for learning models in transliteration extraction.

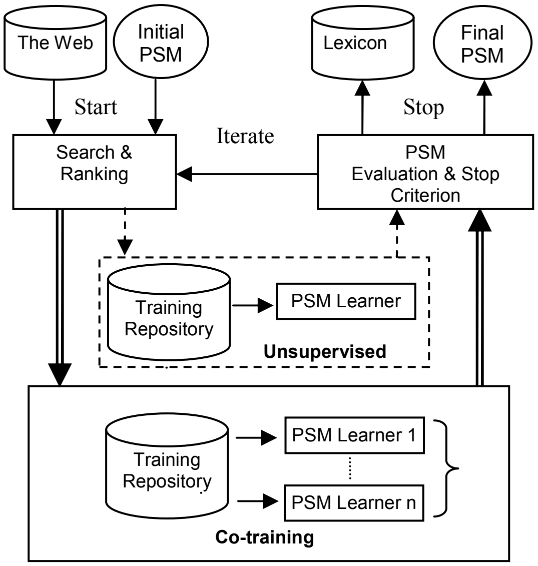

Figure 2. Diagram of unsupervised/multi-view Co-training for transliteration extraction.

To simplify the discussion, here we take a two-view (V_1 and V_2) example to show how Co-training can potentially help. To start with, one can learn a weak hypothesis PSM_1 using V_1 based on a few labeled examples and then apply PSM_1 to all unlabeled examples. If the views are *uncorrelated*, or at least partially *uncorrelated,* these newly labeled examples seen from V_1 augment the training set for V_2. These newly labeled examples present new information from the V_2 point of view, from which one can in turn update the PSM_2. As the views are *compatible,* both V_1 and V_2 label the samples consistently according to the same probabilistic transliteration criteria. In this way, PSMs are boosted each other through such an iterative process between two different views.

```
Given:
  a).  A small set of labeled samples and a set
       of unlabeled samples.
  b).  Two learners A and B are trained on the
       labeled set.

1) Loop for k iterations:
  a).  Learners A and B predict the labels of
       the unlabeled data to augment the labeled
       set;
  b).  Learners A and B are trained on the
       augmented labeled set.
2) Combine models from Learners A and B.
```

Table 1. Co-training with two learners.

Extending the two-view to multi-view, one can develop multiple learners from several subsets of features, each of which approaches the problem from a unique perspective, called a view when taking the Co-training path in Figure 2. Finally, we use outputs from multi-view learners to approximate the manual labeling. The multi-view learning is similar to unsupervised learning in the sense that the learning alleviates the need of labeling and starts with very few labeled data. However, it is also different from the unsupervised learning because the latter does not leverage the natural split of *compatible* and *uncorrelated* features. Two variants of two-view learning strategy can be summarized in Table 1 and Table 2, where the algorithm in Table 1 is referred to as Co-training and the one in Table 2 as Co-EM (Nigam and Ghani. 2000; Muslea et al., 2002).

In Co-training, Learners A and B are trained on the same training data and updated simultaneously. In Co-EM, Learners A and B are trained on labeled set predicted by each other's view, with their models being updated in sequence. In other words, the Co-EM algorithm interchanges the probabilistic labels generated in the view of each other before a new EM iteration. In both cases, the unsupervised, multi-view algorithms use the hypotheses learned to probabilistically label the examples.

```
Given
  a).  A small set of labeled samples and a set of
       unlabeled samples.
  b).  Learner A is trained on a labeled set to
       predict the labels of the unlabeled data.

1) Loop for k iterations
  a).  Learner B is trained on data labeled by
       Learner A to predict the labels of the
       unlabeled data;
  b).  Learner A is trained on data labeled  by
       Learner B to predict the labels of the
       unlabeled data;
2) Combine models from Learners A and B.
```

Table 2. Co-EM with two learners.

The extension of algorithms in Table 1 and 2 to the multi-view transliteration problem is straightforward. After an ensemble of learners are trained, the overall PSM can be expressed as a linear combination of the learners,

$$P(EW \mid CW) = \sum\nolimits_{i=1}^{n} w_i P_i(EW \mid CW), \qquad (8)$$

where w_i is the weight of i^{th} learner $P_i(EW \mid CW)$, which can be learnt by using a development corpus.

5 Experiments

To validate the effectiveness of the learning framework, we conduct a series of experiments in transliteration extraction on a development corpus described later. First, we repeat the experiment in (Kuo et al., 2006) to train a PSM using PSA and GSA feature fusion in a supervised manner, which serves as the upper bound of Co-training or Co-EM system performance. We then train the PSMs with single view V1, V2, V3 and V4 alone in an unsupervised manner. The performance achieved by each view alone can be considered as the baseline for multi-view benchmarking. Then, we run two-view Co-training for different combinations of views on the same development corpus. We expect to see positive effects with the multi-view training. Finally, we run the experiments using two-view Co-training and Co-EM and compare the results.

A 500 MB development corpus is constructed by crawling pages from the Web for the experiments. We first establish a gold standard for performance evaluation by manually labeling the corpus based on the following 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) multiple *E-C* pairs can appear in one sentence; (iii) an *EW* can have multiple valid Chinese transliterations and vice versa.

We first derive 80,094 *E-C* pair candidates from the 500 MB corpus by spotting the co-occurrence of English and Chinese words in the same sentences. This can be done automatically without human intervention. Then, the manual labeling process results in 8,898 qualified *E-C* pairs, also referred to as Distinct Qualified Transliteration Pairs (DQTPs).

To establish comparison, we first train a PSM using all 8,898 DQTPs in a supervised manner and conduct a closed test as reported in Table 3. We further implement three PSM learning strategies and conduct a systematic series of experiments by following the *recognition followed by validation* strategy proposed in (Kuo et al., 2007).

	Precision	Recall	F-measure
Closed test	0.834	0.663	0.739

Table 3. Performance with PSM trained in the supervised manner.

For performance benchmarking, we define the *precision* as the ratio of extracted number of DQTPs over that of total extracted pairs, *recall* as the ratio of extracted number of DQTPs over that of total DQTPs, and *F*-measure as in Eq. (9). They are collectively referred to as extraction performance.

$$F - measure = \frac{2 \times recall \times precision}{recall + precision} \quad (9)$$

5.1 Unsupervised Learning

As formulated in Section 4.1, first, we derive an initial PSM using randomly selected 100 seed DQTPs for each learner and simulate the Web-based learning process: (i) extract *E-C* pairs using the PSM; (ii) add all of the extracted *E-C* pairs to the DQTP pool; (iii) re-estimate the PSM for each view by using the updated DQTP pool. This process is also known as semi-supervised EM (Muslea et al., 2002).

As shown in Figure 3, the unsupervised learning algorithm consistently improves the initial PSM using in all four views. To appreciate the effectiveness of each view, we report the F-measures on each individual view V_1, V_2, V_3 and V_4, as 0.680, 0.620, 0.541 and 0.520, respectively at the 6th iteration. We observe that V_1, the phoneme-based path, achieves the best result.

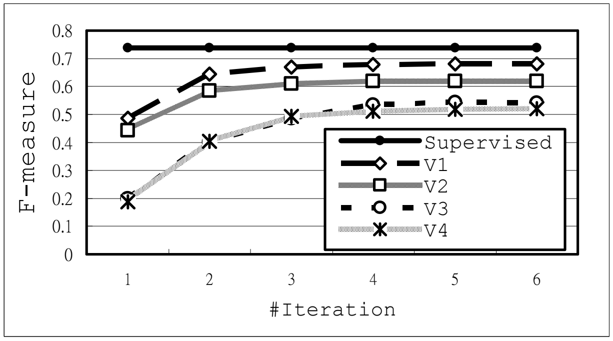

Figure 3. F-measure over iterations using unsupervised learning with individual view.

5.2 Co-training (CT)

We report three typical combinations of two co-working learners or two-view Co-training. Like in unsupervised learning, we start with the same 100 seed DQTPs and an initial PSM model by following the algorithm in Table 1 over 6 iterations.

With two-view Co-training, we obtain 0.726, 0.705, 0.590 and 0.716 in terms of F-measures for V_1+V_2, V_2+V_3, V_3+V_4 and V_1+V_4 at the 6th iteration, as shown in Figure 4. Comparing Figure 3 and 4, we find that Co-training consistently outperforms unsupervised learning by exploiting *compatible* information across different views. The V_1+V_2 Co-training outperforms other Co-training combinations, and surprisingly achieves close performance to that of supervised learning.

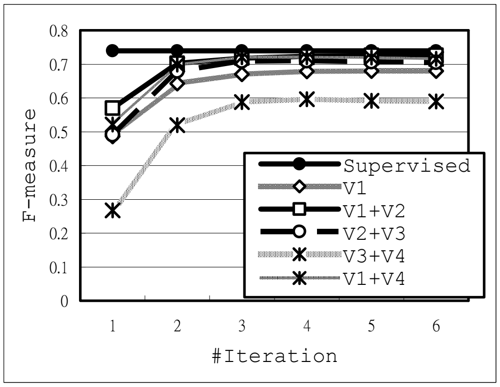

Figure 4. F-measure over iterations using Co-training algorithm

5.3 Co-EM (CE)

Next we start with the same 100 seed DQTPs by initializing the training pool and carry out Co-EM on the same corpus. We build PSM_1 for Learner A and PSM_2 for Learner B. To start with, PSM_1 is learnt from the initial labeled set. We then follow the algorithm in Table 2 by looping in the following two steps over 6 iterations: (i) estimate the PSM_2 from the samples labeled by Learner A (V_1) to extract the high confident *E-C* pairs and augment the DQTP pool with the probabilistically labeled *E-C* pairs; (ii) estimate the PSM_1 from the samples labeled by Learner B (V_2) to extract the high confident *E-C* pairs and augment the DQTP pool with the probabilistically labeled *E-C* pairs. We report the results in Figure 5.

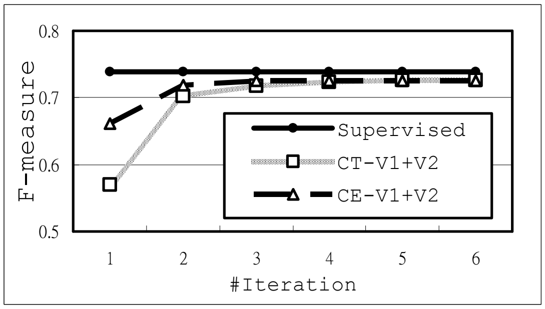

Figure 5. Comparing F-measure over iterations between Co-training (CT) and Co-EM (CE).

To summarize, we compare the performance of six learning methods studied in this paper in Table 4. The Co-training and Co-EM learning approaches have alleviated the need of manual labeling, yet achieving performance close to supervised learning. The multi-view learning effectively leverages multiple *compatible* and partially *uncorrelated* views. It reduces the need of labeled samples from 80,094 to just 100.

We also compare the multi-view learning algorithm with active learning on the same development corpus using same features. We include the results from previously reported work (Kuo et al., 2006) into Table 4 (see Exp. 2) where multiple features are fused in a single active learning process. In Exp. 2, PSA feature is the equivalent of V1 feature in Exp. 4; GSA feature is the equivalent of V4 feature in Exp. 4. In Exp. 4, we carry out V1+V4 two-view Co-training. It is interesting to find that the multi-view learning in this paper achieves better results than active learning in terms of F-measure while reducing the need of manual labeling from 8,191 samples to just 100.

Exp.	Learning algorithm	F-measure	# of samples to label
1	Supervised	0.739	80,094
2	Active Learning (Kuo et al., 2006)	0.710	8,191
3	Unsupervised (V_1)	0.680	100
4	Co-training (V_1+V_4)	0.716	100
5	Co-training (V_1+V_2)	0.726	100
6	Co-EM (V_1+V_2)	0.725	100

Table 4. Comparison of six learning strategies.

6 Conclusions

Fusion of phoneme and grapheme features in transliteration modeling was studied in many previous works. However, it was done through the combination of phoneme and grapheme similarity scores (Bilac and Tanaka, 2004), or by pooling phoneme and grapheme features together into a single-view training process (Oh and Choi, 2006b). This paper presents a new approach that leverages the information across different views to effectively boost the learning from unlabeled data.

We have shown that both Co-training and Co-EM not only outperform the unsupervised learning of single view, but also alleviate the need of data labeling. This reaffirms that multi-view is a viable solution to the learning of transliteration model and hence transliteration extraction. Moving forward, we believe that contextual feature in documents presents another *compatible*, *uncorrelated*, and complementary view to the four views.

We validate the effectiveness of the proposed algorithms by conducting experiments on transliteration extraction. We hope to extend the work further by investigating the possibility of applying the multi-view learning algorithms to machine translation.

References

S. Bilac and H. Tanaka. 2004. Improving back-transliteration by combining information sources, In *Proc. of Int'l Joint Conf. on Natural Language Processing,* pp. 542-547.

S. Blum and T. Mitchell. 1998. Combining Labeled and Unlabeled Data with Co-training, In *Proc. of 11th Conference on Computational Learning Theory*, pp. 92-100.

E. Brill, G. Kacmarcik and C. Brockett. 2001. Automatically Harvesting Katakana-English Term Pairs from Search Engine Query Logs, In *Proc. of Natural Language Processing Pacific Rim Symposium (NLPPRS)*, pp. 393-399.

H.-H. Chen, W.-C. Lin, C.-H. Yang and W.-H. Lin. 2006, Translating-Transliterating Named Entities for Multilingual Information Access, Journal of the American Society for Information Science and Technology, 57(5), pp. 645-659.

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.

K. S. Jeong, S. H. Myaeng, J. S. Lee and K.-S. Choi. 1999. Automatic Identification and Back-transliteration of Foreign Words for Information Retrieval, Information Processing and Management, Vol. 35, pp. 523-540.

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

J.-S. Kuo, H. Li and Y.-K. Yang. 2006. Learning Transliteration Lexicons from the Web, In *Proc. of 44th ACL*, pp. 1129-1136.

J.-S. Kuo, H. Li and Y.-K. Yang. 2007. A Phonetic Similarity Model for Automatic Extraction of Transliteration Pairs, ACM Transactions on Asian Language Information Processing. 6(2), pp. 1-24.

J.-S. Lee. 1999. An English-Korean Transliteration and Retransliteration Model for Cross-Lingual Information Retrieval, PhD Thesis, Department of Computer Science, KAIST.

D. D. Lewis and J. Catlett. 1994. Heterogeneous Uncertainty Sampling for Supervised Learning, In *Proc. of Int'l Conference on Machine Learning (ICML)*, 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.

C. D. Manning and H. Scheutze. 1999. Fundamentals of Statistical Natural Language Processing, The MIT Press.

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 *Proceedings of Automatic Speech Recognition Understanding (ASRU)*, pp. 311-314.

I. Muslea, S. Minton and C. A. Knoblock. 2002. Active + Semi-supervised learning = Robust Multi-View Learning, In *Proc. of the 9th Int'l Conference on Machine Learning*, pp. 435-442.

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.

K. Nigam and R. Ghani. 2000. Analyzing the Effectiveness and Applicability of Co-training, In *Proc. of the 9th Conference in Information and Knowledge and Management*, pp. 86-93.

J.-H. Oh, K.-S. Choi and H. Isahara. 2006a. A Machine Transliteration Model based on Graphemes and Phonemes, ACM TALIP, Vol. 5, No. 3, pp. 185-208.

J.-H. Oh and K.-S. Choi. 2006b. An Ensemble of Transliteration Models for Information Retrieval, In Information Processing and Management, Vol. 42, pp. 980-1002.

R. Sproat, T. Tao and C. Zhai. 2006. Named Entity Transliteration with Comparable Corpora, In *Proc. of 44th ACL,* pp. 73-80.

G. Tür, D. Hakkani-Tür and R. E. Schapire. 2005. Combining Active and Semi-supervised Learning for Spoken Language Understanding, Speech Communication, 45, pp. 171-186.

B. Vauqois. 1988. A Survey of Formal Grammars and Algorithms for Recognition and Transformation in Machine Translation, IFIP Congress-68, reprinted TAO: Vingtcinq Ans de Traduction Automatique - Analectes in C. Boitet, Ed., Association Champollin, Grenoble, pp.201-213

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