A Statistical Model for Unsupervised and Semi-supervised Transliteration Mining

Hassan Sajjad **Alexander Fraser** **Helmut Schmid**
Institute for Natural Language Processing
University of Stuttgart
{sajjad,fraser,schmid}@ims.uni-stuttgart.de

Abstract

We propose a novel model to automatically extract transliteration pairs from parallel corpora. Our model is efficient, language pair independent and mines transliteration pairs in a consistent fashion in both unsupervised and semi-supervised settings. We model transliteration mining as an interpolation of transliteration and non-transliteration sub-models. We evaluate on NEWS 2010 shared task data and on parallel corpora with competitive results.

1 Introduction

Transliteration mining is the extraction of transliteration pairs from unlabelled data. Most transliteration mining systems are built using labelled training data or using heuristics to extract transliteration pairs. These systems are language pair dependent or require labelled information for training. Our system extracts transliteration pairs in an unsupervised fashion. It is also able to utilize labelled information if available, obtaining improved performance.

We present a novel model of transliteration mining defined as a mixture of a transliteration model and a non-transliteration model. The transliteration model is a joint source channel model (Li et al., 2004). The non-transliteration model assumes no correlation between source and target word characters, and independently generates a source and a target word using two fixed unigram character models. We use Expectation Maximization (EM) to learn parameters maximizing the likelihood of the interpolation of both sub-models. At test time, we label word pairs as transliterations if they have a higher probability assigned by the transliteration sub-model than by the non-transliteration sub-model.

We extend the unsupervised system to a semi-supervised system by adding a new S-step to the EM algorithm. The S-step takes the probability estimates from unlabelled data (computed in the M-step) and uses them as a backoff distribution to smooth probabilities which were estimated from labelled data. The smoothed probabilities are then used in the next E-step. In this way, the parameters learned by EM are constrained to values which are close to those estimated from the labelled data.

We evaluate our unsupervised and semi-supervised transliteration mining system on the datasets available from the NEWS 2010 shared task on transliteration mining (Kumaran et al., 2010b). We call this task *NEWS10* later on. Compared with a baseline unsupervised system our unsupervised system achieves up to 5% better F-measure. On the NEWS10 dataset, our unsupervised system achieves an F-measure of up to 95.7%, and on three language pairs, it performs better than all systems which participated in NEWS10. We also evaluate our semi-supervised system which additionally uses the NEWS10 labelled data for training. It achieves an improvement of up to 3.7% F-measure over our unsupervised system. Additional experiments on parallel corpora show that we are able to effectively mine transliteration pairs from very noisy data.

The paper is organized as follows. Section 2 describes previous work. Sections 3 and 4 define our unsupervised and semi-supervised models. Section 5 presents the evaluation. Section 6 concludes.

2 Previous Work

We first discuss the literature on semi-supervised and supervised techniques for transliteration mining and then describe a previously defined unsupervised system. Supervised and semi-supervised systems use a manually labelled set of training data to learn character mappings between source and target strings. The labelled training data either consists of a few hundred transliteration pairs or of just a few carefully selected transliteration pairs. The NEWS 2010 shared task on transliteration mining (NEWS10) (Kumaran et al., 2010b) is a semi-supervised task conducted on Wikipedia InterLanguage Links (WIL) data. The NEWS10 dataset contains 1000 labelled examples (called the "seed data") for initial training. All systems which participated in the NEWS10 shared task are either supervised or semi-supervised. They are described in (Kumaran et al., 2010a). Our transliteration mining model can mine transliterations without using any labelled data. However, if there is some labelled data available, our system is able to use it effectively.

The transliteration mining systems evaluated on the NEWS10 dataset generally used heuristic methods, discriminative models or generative models for transliteration mining (Kumaran et al., 2010a).

The heuristic-based system of Jiampojamarn et al. (2010) is based on the edit distance method which scores the similarity between source and target words. They presented two discriminative methods – an SVM-based classifier and alignment-based string similarity for transliteration mining. These methods model the conditional probability distribution and require supervised/semi-supervised information for learning. We propose a flexible generative model for transliteration mining usable for both unsupervised and semi-supervised learning.

Previous work on generative approaches uses Hidden Markov Models (Nabende, 2010; Darwish, 2010; Jiampojamarn et al., 2010), Finite State Automata (Noeman and Madkour, 2010) and Bayesian learning (Kahki et al., 2011) to learn transliteration pairs from labelled data. Our method is different from theirs as our generative story explains the unlabelled data using a combination of a transliteration and a non-transliteration sub-model. The transliteration model jointly generates source and target strings, whereas the non-transliteration system generates them independently of each other.

Sajjad et al. (2011) proposed a heuristic-based unsupervised transliteration mining system. We later call it *Sajjad11*. It is the only unsupervised mining system that was evaluated on the NEWS10 dataset up until now, as far as we know. That system is computationally expensive. We show in Section 5 that its runtime is much higher than that of our system.

In this paper, we propose a novel model-based approach to transliteration mining. Our approach is language pair independent – at least for alphabetic languages – and efficient. Unlike the previous unsupervised system, and unlike the supervised and semi-supervised systems we mentioned, our model can be used for both unsupervised and semi-supervised mining in a consistent way.

3 Unsupervised Transliteration Mining Model

A source word and its corresponding target word can be character-aligned in many ways. We refer to a possible alignment sequence which aligns a source word $\mathbf{e}$ and a target word $\mathbf{f}$ as *"a"*. The function $Align(e, f)$ returns the set of all valid alignment sequences a of a word pair (e, f). The joint transliteration probability $p_1(e, f)$ of a word pair is the sum of the probabilities of all alignment sequences:

$$p_1(e, f) = \sum_{a \in Align(e,f)} p(a) \qquad (1)$$

Transliteration systems are trained on a list of transliteration pairs. The alignment between the transliteration pairs is learned with Expectation Maximization (EM). We use a simple unigram model, so an alignment sequence from function $Align(e, f)$ is a combination of 0–1, 1–1, and 1–0 character alignments between a source word $\mathbf{e}$ and its transliteration $\mathbf{f}$. We refer to a character alignment unit as *"multigram"* later on and represent it by the symbol *"q"*. A sequence of multigrams forms an alignment of a source and target word. The probability of a sequence of multigrams a is the product of the probabilities of the multigrams it contains.

$$p(a) = p(q_1, q_2, ..., q_{|a|}) = \prod_{j=1}^{|a|} p(q_j) \qquad (2)$$

While transliteration systems are trained on a clean list of transliteration pairs, our transliteration mining system has to learn from data containing both transliterations and non-transliterations. The transliteration model $p_1(e, f)$ handles only the transliteration pairs. We propose a second model $p_2(e, f)$ to deal with non-transliteration pairs (the *"non-transliteration model"*). Interpolation with the non-transliteration model allows the transliteration model to concentrate on modelling transliterations during EM training. After EM training, transliteration word pairs are assigned a high probability by the transliteration submodel and a low probability by the non-transliteration submodel, and vice versa for non-transliteration pairs. This property is exploited to identify transliterations.

In a non-transliteration word pair, the characters of the source and target words are unrelated. We model them as randomly seeing a source word and a target word together. The non-transliteration model uses random generation of characters from two unigram models. It is defined as follows:

$$p_2(e, f) = p_E(e)\ p_F(f) \tag{3}$$

$p_E(e) = \prod_{i=1}^{|e|} p_E(e_i)$ and $p_F(f) = \prod_{i=1}^{|f|} p_F(f_i)$.

The transliteration mining model is an interpolation of the transliteration model $p_1(e, f)$ and the non-transliteration model $p_2(e, f)$:

$$p(e, f) = (1 - \lambda)p_1(e, f) + \lambda p_2(e, f) \tag{4}$$

λ is the prior probability of non-transliteration.

3.1 Model Estimation

In this section, we discuss the estimation of the parameters of the transliteration model $p_1(e, f)$ and the non-transliteration model $p_2(e, f)$.

The non-transliteration model consists of two unigram character models. Their parameters are estimated from the source and target words of the training data, respectively, and the parameters do not change during EM training.

For the transliteration model, we implement a simplified form of the grapheme-to-phoneme converter, g2p (Bisani and Ney, 2008). In the following, we use notations from Bisani and Ney (2008). g2p learns m-to-n character alignments between a source and a target word. We restrict ourselves to 0–1,1–1,1–0 character alignments and to a unigram model.[1] The Expectation Maximization (EM) algorithm is used to train the model. It maximizes the likelihood of the training data. In the E-step the EM algorithm computes expected counts for the multigrams and in the M-step the multigram probabilities are reestimated from these counts. These two steps are iterated. For the first EM iteration, the multigram probabilities are initialized with a uniform distribution and λ is set to 0.5.

The expected count of a multigram q (E-step) is computed by multiplying the posterior probability of each alignment a with the frequency of q in a and summing these weighted frequencies over all alignments of all word pairs.

$$c(q) \quad = \quad \sum_{i=1}^{N} \sum_{a \in Align(e_i, f_i)} \frac{(1 - \lambda)p_1(a, e_i, f_i)}{p(e_i, f_i)} n_q(a)$$

$n_q(a)$ is here the number of times the multigram q occurs in the sequence a and $p(e_i, f_i)$ is defined in Equation 4. The new estimate of the probability of a multigram is given by:

$$p(q) = \frac{c(q)}{\sum_{q'} c(q')} \tag{5}$$

Likewise, we calculate the expected count of non-transliterations by summing the posterior probabilities of non-transliteration given each word pair:

$$c_{ntr} = \sum_{i=1}^{N} p_{ntr}(e_i, f_i) = \sum_{i=1}^{N} \frac{\lambda p_2(e_i, f_i)}{p(e_i, f_i)} \tag{6}$$

λ is then reestimated by dividing the expected count of non-transliterations by N.

3.2 Implementation Details

We use the Forward-Backward algorithm to estimate the counts of multigrams. The algorithm has a forward variable α and a backward variable β which are calculated in the standard way (Deligne and Bimbot, 1995). Consider a node r which is connected with a node s via an arc labelled with the multigram q. The expected count of a transition between r and s is calculated using the forward and backward probabilities as follows:

$$\gamma'_{rs} = \frac{\alpha(r)\ p(q)\ \beta(s)}{\alpha(E)} \tag{7}$$

[1] In preliminary experiments, using an n-gram order of greater than one or more than one character on the source side or the target side or both sides of the multigram caused the transliteration model to incorrectly learn non-transliteration information from the training data.

where E is the final node of the graph.

We multiply the expected count of a transition by the posterior probability of transliteration $(1 - p_{ntr}(e, f))$ which indicates how likely the string pair is to be a transliteration. The counts γ_{rs} are then summed for all multigram types q over all training pairs to obtain the frequencies $c(q)$ which are used to reestimate the multigram probabilities according to Equation 5.

4 Semi-supervised Transliteration Mining Model

Our unsupervised transliteration mining system can be applied to language pairs for which no labelled data is available. However, the unsupervised system is focused on high recall and also mines close transliterations (see Section 5 for details). In a task dependent scenario, it is difficult for the unsupervised system to mine transliteration pairs according to the details of a particular definition of what is considered a transliteration (which may vary somewhat with the task). In this section, we propose an extension of our unsupervised model which overcomes this shortcoming by using labelled data. The idea is to rely on probabilities from labelled data where they can be estimated reliably and to use probabilities from unlabelled data where the labelled data is sparse. This is achieved by smoothing the labelled data probabilities using the unlabelled data probabilities as a backoff.

4.1 Model Estimation

We calculate the unlabelled data probabilities in the E-step using Equation 4. For labelled data (containing only transliterations) we set $\lambda = 0$ and get:

$$p(e, f) = \sum_{a \in Align(e,f)} p_1(e, f, a) \qquad (8)$$

In every EM iteration, we smooth the probability distribution in such a way that the estimates of the multigrams of the unlabelled data that do not occur in the labelled data would be penalized. We obtain this effect by smoothing the probability distribution of unlabelled and labelled data using a technique similar to Witten-Bell smoothing (Witten and Bell, 1991), as we describe below.

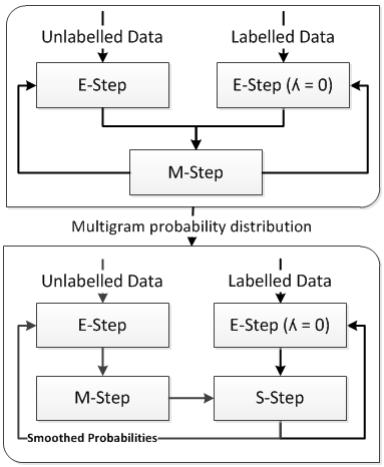

Figure 1: Semi-supervised training

4.2 Implementation Details

We divide the training process of semi-supervised mining in two steps as shown in Figure 1. The first step creates a reasonable alignment of the labelled data from which multigram counts can be obtained. The labelled data is a small list of transliteration pairs. Therefore we use the unlabelled data to help correctly align it and train our unsupervised mining system on the combined labelled and unlabelled training data. In the expectation step, the prior probability of non-transliteration λ is set to zero on the labelled data since it contains only transliterations. The first step passes the resulting multigram probability distribution to the second step.

We start the second step with the probability estimates from the first step and run the E-step separately on labelled and unlabelled data. The E-step on the labelled data is done using Equation 8, which forces the posterior probability of non-transliteration to zero, while the E-step on the unlabelled data uses Equation 4. After the two E-steps, we estimate a probability distribution from the counts obtained from the unlabelled data (M-step) and use it as a backoff distribution in computing smoothed probabilities from the labelled data counts (S-step).

The smoothed probability estimate $\hat{p}(q)$ is:

$$\hat{p}(q) = \frac{c_s(q) + \eta_s p(q)}{N_s + \eta_s} \qquad (9)$$

where $c_s(q)$ is the labelled data count of the multigram q, $p(q)$ is the unlabelled data probability estimate, and $N_s = \sum_q c_s(q)$, and η_s is the number of different multigram types observed in the Viterbi alignment of the labelled data.

5 Evaluation

We evaluate our unsupervised system and semi-supervised system on two tasks, NEWS10 and parallel corpora. NEWS10 is a standard task on transliteration mining from WIL. On NEWS10, we compare our results with the unsupervised mining system of Sajjad et al. (2011), the best supervised and semi-supervised systems presented at NEWS10 (Kumaran et al., 2010b) and the best supervised and semi-supervised results reported in the literature for the NEWS10 task. For the challenging task of mining from parallel corpora, we use the English/Hindi and English/Arabic gold standard provided by Sajjad et al. (2011) to evaluate our results.

5.1 Experiments using the NEWS10 Dataset

We conduct experiments on four language pairs: English/Arabic, English/Hindi, English/Tamil and English/Russian using data provided at NEWS10. Every dataset contains training data, seed data and reference data. The NEWS10 data consists of pairs of titles of the same Wikipedia pages written in different languages, which may be transliterations or translations. The seed data is a list of 1000 transliteration pairs provided to semi-supervised systems for initial training. We use the seed data only in our semi-supervised system, and not in the unsupervised system. The reference data is a small subset of the training data which is manually annotated with positive and negative examples.

5.1.1 Training

We word-aligned the parallel phrases of the training data using GIZA++ (Och and Ney, 2003), and symmetrized the alignments using the grow-diag-final-and heuristic (Koehn et al., 2003). We extract all word pairs which occur as 1-to-1 alignments (like Sajjad et al. (2011)) and later refer to them as the *word-aligned list*. We compared the word-aligned list with the NEWS10 reference data and found that the word-aligned list is missing some transliteration pairs because of word-alignment errors. We built another list by adding a word pair for every source word that cooccurs with a target word in a parallel phrase/sentence and call it the *cross-product list* later on. The cross-product list is noisier but contains almost all transliteration pairs in the corpus.

	Word-aligned			Cross-product		
	P	R	F	P	R	F
EA	27.8	97.1	43.3	14.3	98.0	25.0
EH	42.5	98.7	59.4	20.5	99.6	34.1
ET	32.0	98.1	48.3	17.2	99.6	29.3
ER	25.5	95.6	40.3	12.8	99.0	22.7

Table 1: Statistics of word-aligned and cross-product list calculated from the NEWS10 dataset, before mining. *EA* is English/Arabic, *EH* is English/Hindi, *ET* is English/Tamil and *ER* is English/Russian

Table 1 shows the statistics of the word-aligned list and the cross-product list calculated using the NEWS10 reference data.[2] The word-aligned list calculated from the NEWS10 dataset is used to compare our unsupervised system with the unsupervised system of Sajjad et al. (2011) on the same training data. All the other experiments on NEWS10 use cross-product lists. We remove numbers from both lists as they are defined as non-transliterations (Kumaran et al., 2010b).

5.1.2 Unsupervised Transliteration Mining

We run our unsupervised transliteration mining system on the word-aligned list and the cross-product list. The word pairs with a posterior probability of transliteration $1 - p_{ntr}(e, f) = 1 - \lambda p_2(e_i, f_i)/p(e_i, f_i)$ greater than 0.5 are selected as transliteration pairs.

We compare our unsupervised system with the unsupervised system of Sajjad11. Our unsupervised system trained on the word-aligned list shows F-measures of 91.7%, 95.5%, 92.9% and 77.7% which is 4.3%, 3.3%, 2.8% and 1.7% better than the system of Sajjad11 on English/Arabic, English/Hindi, English/Tamil and English/Russian respectively.

Sajjad11 is computationally expensive. For instance, a phrase-based statistical MT system is built once in every iteration of the heuristic procedure. We ran Sajjad11 on the English/Russian word-aligned list using a 2.4 GHz Dual-Core AMD machine, which took almost 10 days. On the same machine, our transliteration mining system only takes 1.5 hours to finish the same experiment.

[2]Due to inconsistent word definition used in the reference data, we did not achieve 100% recall in our cross-product list. For example, the underscore is defined as a word boundary for English WIL phrases. This assumption is not followed for certain phrases like "New_York" and "New_Mexico".

	Unsupervised		Semi-supervised/Supervised			
	SJD	O_U	O_S	S_{Best}	GR	DBN
EA	87.4	92.4	92.7	91.5	94.1	-
EH	92.2	95.7	96.3	94.4	93.2	95.5
ET	90.1	93.2	94.6	91.4	95.5	93.9
ER	76.0	79.4	83.1	87.5	92.3	82.5

Table 2: F-measure results on NEWS10 datasets where SJD is the unsupervised system of Sajjad11, O_U is our unsupervised system built on the cross-product list, O_S is our semi-supervised system, S_{Best} is the best NEWS10 system, GR is the supervised system of Kahki et al. (2011) and DBN is the semi-supervised system of Nabende (2011)

Our unsupervised mining system built on the cross-product list consistently outperforms the one built on the word-aligned list. Later, we consider only the system built on the cross-product list. Table 2 shows the results of our unsupervised system O_U in comparison with the unsupervised system of Sajjad11 (SJD), the best semi-supervised systems presented at NEWS10 (S_{BEST}) and the best semi-supervised results reported on the NEWS10 dataset (GR, DBN). On three language pairs, our unsupervised system performs better than all semi-supervised systems which participated in NEWS10. It has competitive results with the best supervised results reported on NEWS10 datasets. On English/Hindi, our unsupervised system outperforms the state-of-the-art supervised and semi-supervised systems. Kahki et al. (2011) (GR) achieved the best results on English/Arabic, English/Tamil and English/Russian. For the English/Arabic task, they normalized the data using language dependent heuristics[3] and also used a non-standard evaluation method (discussed in Section 5.1.4).

On the English/Russian dataset, our unsupervised system faces the problem that it extracts cognates as transliterations. The same problem was reported in Sajjad et al. (2011). Cognates are close transliterations which differ by only one or two characters from an exact transliteration pair. The unsupervised system learns to delete the additional one or two characters with a high probability and incorrectly mines such word pairs as transliterations.

[3]They applied an Arabic word segmenter which uses language dependent information. Arabic long vowels which have identical sound but are written differently were merged to one form. English characters were normalized by dropping accents.

	Unsupervised			Semi-supervised		
	P	R	F	P	R	F
EA	89.2	95.7	92.4	92.9	92.4	92.7
EH	92.6	99.0	95.7	95.5	97.0	96.3
ET	88.3	98.6	93.2	93.4	95.8	94.6
ER	67.2	97.1	79.4	74.0	94.9	83.1

Table 3: Precision(P), Recall(R) and F-measure(F) of our unsupervised and semi-supervised transliteration mining systems on NEWS10 datasets

5.1.3 Semi-supervised Transliteration Mining

Our semi-supervised system uses similar initialization of the parameters as used for unsupervised system. Table 2 shows on three language pairs, our semi-supervised system O_S only achieves a small gain in F-measure over our unsupervised system O_U. This shows that the unlabelled training data is already providing most of the transliteration information. The seed data is used to help the transliteration mining system to learn the right definition of transliteration. On the English/Russian dataset, our semi-supervised system achieves almost 7% increase in precision with a 2.2% drop in recall compared to our unsupervised system. This provides a 3.7% gain on F-measure. The increase in precision shows that the seed data is helping the system in disambiguating transliteration pairs from cognates.

5.1.4 Discussion

The unsupervised system produces lists with high recall. The semi-supervised system tends to better balance out precision and recall. Table 3 compares the precision, recall and F-measure of our unsupervised and semi-supervised mining systems.

The errors made by our semi-supervised system can be classified into the following categories:

Pronunciation differences: English proper names may be pronounced differently in other languages. Sometimes, English short vowels are converted to long vowels in Hindi such as the English word "Lanthanum" which is pronounced "Laanthanum" in Hindi. Our transliteration mining system wrongly extracts such pairs as transliterations.

In some cases, different vowels are used in two languages. The English word "January" is pronounced as "Janvary" in Hindi. Such word pairs are non-transliterations according to the gold standard but our system extracts them as transliterations. Ta-

English	Hindi	English	Hindi
Lanthanum	लाञ्थनम/Laanthanum	Sailendra	शैलेन्द्र/Shalendra
January	जनवरी/Janvary	August	अगस्त/Aagast

Table 4: Word pairs with pronunciation differences

English	Arabic	English	Arabic
Basrah	البصرة/Albasrah	Nasr	النصر/Alnasr
Kuwait	الكويت/Alkuwait	Riyadh	الرياض/Alriyadh

Table 5: Examples of word pairs which are wrongly annotated as transliterations in the gold standard

ble 4 shows a few examples of such word pairs.

Inconsistencies in the gold standard: There are several inconsistencies in the gold standard where our transliteration system correctly identifies a word pair as a transliteration but it is marked as a non-transliteration or vice versa. Consider the example of the English word "George" which is pronounced as "Jaarj" in Hindi. Our semi-supervised system learns this as a non-transliteration but it is wrongly annotated as a transliteration in the gold standard.

Arabic nouns have an article "al" attached to them which is translated in English as "the". There are various cases in the training data where an English noun such as "Quran" is matched with an Arabic noun "alQuran". Our mining system classifies such cases as non-transliterations, but 24 of them are incorrectly annotated as transliterations in the gold standard. We did not correct this, and are therefore penalized. Kahki et al. (2011) preprocessed such Arabic words and separated "al" from the noun "Quran" before mining. They report a match if the version of the Arabic word with "al" appears with the corresponding English word in the gold standard. Table 5 shows examples of word pairs which are wrongly annotated as transliterations.

Cognates: Sometimes a word pair differs by only one or two ending characters from a true transliteration. For example in the English/Russian training data, the Russian nouns are marked with cases whereas their English counterparts do not mark the case or translate it as a separate word. Often the Russian word differs only by the last character from a correct transliteration of the English word. Due to the large amount of such word pairs in the English/Russian data, our mining system learns to delete the final case marking characters from the Russian words. It assigns a high transliteration probability to these word pairs and extracts them as transliterations. Table 6 shows some examples.

English	Russian	English	Russian
Studio	Студия/Studiya	Catalonia	Каталонии/Katalonii
Estonia	Эстонии/Estonii	Geography	География/Geografiya

Table 6: A few examples of English/Russian cognates

There are two English/Russian supervised systems which are better than our semi-supervised system. The Kahki et al. (2011) system is built on seed data only. Jiampojamarn et al. (2010)'s best system on English/Russian is based on the edit distance method. Both of these systems are focused on high precision. Our semi-supervised system is focused on high recall at the cost of lower precision.[4]

5.2 Transliteration Mining using Parallel Corpora

The percentage of transliteration pairs in the NEWS10 datasets is high. We further check the effectiveness of our unsupervised and semi-supervised mining systems by evaluating them on parallel corpora with as few as 2% transliteration pairs.

We conduct experiments using two language pairs, English/Hindi and English/Arabic. The English/Hindi corpus is from the shared task on word alignment organized as part of the ACL 2005 Workshop on Building and Using Parallel Texts (WA05) (Martin et al., 2005). For English/Arabic, we use 200,000 parallel sentences from the United Nations (UN) corpus (Eisele and Chen, 2010). The English/Hindi and English/Arabic transliteration gold standards were provided by Sajjad et al. (2011).

5.2.1 Experiments

We follow the procedure for creating the training data described in Section 5.1.1 and build a word-aligned list and a cross-product list from the parallel corpus. We first train and test our unsupervised mining system on the word-aligned list and compare our results with Sajjad et al. Table 7 shows the results. Our unsupervised system achieves 0.6% and 1.8% higher F-measure than Sajjad et al. respectively.

The cross-product list is huge in comparison to the word-aligned list. It is noisier than the word-

[4]We implemented a bigram version of our system to learn the contextual information at the end of the word pairs, but only achieved a gain of less than 1% F-measure over our unigram semi-supervised system. Details are omitted due to space.

	TP	FN	TN	FP	P	R	F
EH_{SJD}	170	10	2039	45	79.1	94.4	86.1
EH_O	176	4	2034	50	77.9	97.8	86.7
EA_{SJD}	197	91	6580	59	77.0	68.4	72.5
EA_O	288	0	6440	199	59.1	100	74.3

Table 7: Transliteration mining results of our unsupervised system and Sajjad11 system trained and tested on the word-aligned list of English/Hindi and English/Arabic parallel corpus

	TP	FN	TN	FP	P	R	F
EH_U	393	19	12279	129	75.3	95.4	84.2
EH_S	365	47	12340	68	84.3	88.6	86.4
EA_U	277	11	6444	195	58.7	96.2	72.9
EA_S	272	16	6497	142	65.7	94.4	77.5

Table 8: Transliteration mining results of our unsupervised and semi-supervised systems trained on the word-aligned list and tested on the cross-product list of English/Hindi and English/Arabic parallel corpus

aligned list but has almost 100% recall of transliteration pairs. The English-Hindi cross-product list has almost 55% more transliteration pairs (412 types) than the word-aligned list (180 types). We can not report these numbers on the English/Arabic cross-product list since the English/Arabic gold standard is built on the word-aligned list.

In order to keep the experiment computationally inexpensive, we train our mining systems on the word-aligned list and test them on the cross-product list.[5] We also perform the first semi-supervised evaluation on this task. For our semi-supervised system, we additionally use the English/Hindi and English/Arabic seed data provided by NEWS10.

Table 8 shows the results of our unsupervised and semi-supervised systems on the English/Hindi and English/Arabic parallel corpora. Our unsupervised system achieves higher recall than our semi-supervised system but lower precision. The semi-supervised system shows an improvement in F-measure for both language pairs. We looked into the errors made by our systems. The mined transliteration pairs of our unsupervised system contains 65 and 111 close transliterations for the English/Hindi and English/Arabic task respectively. The close transliterations only differ by one or two characters from correct transliterations. We think these pairs provide transliteration information to the systems and help them to avoid problems with data sparseness. Our semi-supervised system uses the seed data to identify close transliterations as non-transliterations and decreases the number of false positives. They are reduced to 35 and 89 for English/Hindi and English/Arabic respectively. The seed data and the training data used in the semi-supervised system are from different domains (Wikipedia and UN). Seed data extracted from the same domain is likely to work better, resulting in even higher scores than we have reported.

[5]There are some multigrams of the cross-product list which are unknown to the model learned on the word-aligned list. We define their probability as the inverse of the number of multigram tokens in the Viterbi alignment of the labelled and unlabelled data together.

6 Conclusion and Future Work

We presented a novel model to automatically mine transliteration pairs. Our approach is efficient and language pair independent (for alphabetic languages). Both the unsupervised and semi-supervised systems achieve higher accuracy than the only unsupervised transliteration mining system we are aware of and are competitive with the state-of-the-art supervised and semi-supervised systems. Our semi-supervised system outperformed our unsupervised system, in particular in the presence of prevalent cognates in the Russian/English data.

In future work, we plan to adapt our approach to language pairs where one language is alphabetic and the other language is non-alphabetic such as English/Japanese. These language pairs require one-to-many character mappings to learn transliteration units, while our current system only learns unigram character alignments.

Acknowledgments

The authors wish to thank the anonymous reviewers. We would like to thank Syed Aoun Raza for discussions of implementation efficiency. Hassan Sajjad was funded by the Higher Education Commission of Pakistan. Alexander Fraser was funded by Deutsche Forschungsgemeinschaft grant Models of Morphosyntax for Statistical Machine Translation. Helmut Schmid was supported by Deutsche Forschungsgemeinschaft grant SFB 732. This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

References

Maximilian Bisani and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5).

Kareem Darwish. 2010. Transliteration mining with phonetic conflation and iterative training. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.

Sabine Deligne and Frédéric Bimbot. 1995. Language modeling by variable length sequences : Theoretical formulation and evaluation of multigrams. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, Los Alamitos, CA, USA.

Andreas Eisele and Yu Chen. 2010. MultiUN: A multilingual corpus from United Nation documents. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.

Ali El Kahki, Kareem Darwish, Ahmed Saad El Din, Mohamed Abd El-Wahab, Ahmed Hefny, and Waleed Ammar. 2011. Improved transliteration mining using graph reinforcement. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Edinburgh, UK.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, Edmonton, Canada.

A Kumaran, Mitesh M. Khapra, and Haizhou Li. 2010a. Report of NEWS 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.

A Kumaran, Mitesh M. Khapra, and Haizhou Li. 2010b. Whitepaper of NEWS 2010 shared task on transliteration mining. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.

Haizhou Li, Zhang Min, and Su Jian. 2004. A joint source-channel model for machine transliteration. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Barcelona, Spain.

Joel Martin, Rada Mihalcea, and Ted Pedersen. 2005. Word alignment for languages with scarce resources. In *ParaText '05: Proceedings of the ACL Workshop on Building and Using Parallel Texts*, Morristown, NJ, USA.

Peter Nabende. 2010. Mining transliterations from wikipedia using pair hmms. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.

Peter Nabende. 2011. Mining transliterations from Wikipedia using dynamic bayesian networks. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, Hissar, Bulgaria.

Sara Noeman and Amgad Madkour. 2010. Language independent transliteration mining system using finite state automata framework. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.

Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2011. An algorithm for unsupervised transliteration mining with an application to word alignment. In *Proceedings of the 49th Annual Conference of the Association for Computational Linguistics*, Portland, USA.

Ian H. Witten and Timothy C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. In *IEEE Transactions on Information Theory*, volume 37.