Statistical Models for Unsupervised, Semi-Supervised, and Supervised Transliteration Mining

Hassan Sajjad*
Qatar Computing Research Institute

Helmut Schmid
Ludwig Maximilian University of Munich

Alexander Fraser
Ludwig Maximilian University of Munich

Hinrich Schütze
Ludwig Maximilian University of Munich

We present a generative model that efficiently mines transliteration pairs in a consistent fashion in three different settings: unsupervised, semi-supervised, and supervised transliteration mining. The model interpolates two sub-models, one for the generation of transliteration pairs and one for the generation of non-transliteration pairs (i.e., noise). The model is trained on noisy unlabeled data using the EM algorithm. During training the transliteration sub-model learns to generate transliteration pairs and the fixed non-transliteration model generates the noise pairs. After training, the unlabeled data is disambiguated based on the posterior probabilities of the two sub-models. We evaluate our transliteration mining system on data from a transliteration mining shared task and on parallel corpora. For three out of four language pairs, our system outperforms all semi-supervised and supervised systems that participated in the NEWS 2010 shared task. On word pairs extracted from parallel corpora with fewer than 2% transliteration pairs, our system achieves up to 86.7% F-measure with 77.9% precision and 97.8% recall.

* Much of the research presented here was conducted while the authors were at the University of Stuttgart, Germany. E-mail: hsajjad@gf.org.qa; {schmid,fraser}@cis.uni-muenchen.de

Submission received: 17 November 2015; revised version received: 29 June 2016; accepted for publication: 7 July 2016.

doi:10.1162/COLI_a_00286

1. Introduction

Transliteration converts a word from a source script into a target script. The English words Alberto and Doppler, for example, can be written in Arabic script as البرتو/albrtw and دوبلر/dwblr, respectively, and are examples of transliteration.

Automatic transliteration is useful in many NLP applications such as cross-language information retrieval, statistical machine translation, building of comparable corpora, terminology extraction, and so forth. Most transliteration systems are trained on a list of **transliteration pairs** which consist of a word and its transliteration. However, manually labeled transliteration pairs are only available for a few language pairs. Therefore it is attractive to extract transliteration pairs automatically from a noisy list of transliteration candidates, which can be obtained from aligned bilingual corpora, for instance. This extraction process is called **transliteration mining**.

There are rule-based, supervised, semi-supervised, and unsupervised ways to mine transliteration pairs. **Rule-based methods** apply weighted handwritten rules that map characters between two languages, and compute a weighted edit distance metric that assigns a score to every candidate word pair. Pairs with an edit distance below a given threshold are extracted (Jiampojamarn et al. 2010; Noeman and Madkour 2010; Sajjad et al. 2011). **Supervised transliteration mining systems** (Nabende 2010; Noeman and Madkour 2010; El-Kahki et al. 2011) make use of an initial list of transliteration pairs that is automatically aligned at the character level. The systems are trained on the aligned data and applied to an unlabeled list of candidate word pairs. Word pairs with a probability greater than a certain threshold are classified as transliteration pairs. Similarly to supervised approaches, **semi-supervised systems** (Sherif and Kondrak 2007; Darwish 2010) also use a list of transliteration pairs for training. However, here the list is generally small. The systems thus do not solely rely on it to mine transliteration pairs. They use both the list of transliteration pairs and unlabeled data for training.

We are only aware of two **unsupervised systems** (requiring no labeled data). One of them was proposed by Fei Huang (2005). He extracts named entity pairs from a bilingual corpus, converts all words into Latin script by romanization, and classifies them into transliterations and non-transliterations based on the edit distance. This system still requires a named entity tagger to generate the candidate pairs, a list of mapping rules to convert non-Latin scripts to Latin, and labeled data to optimize parameters. The only previous system that requires no such resources is that of Sajjad, Fraser, and Schmid (2011). They extract transliteration pairs by iteratively filtering a list of candidate pairs. The downsides of their method are inefficiency and inflexibility. It requires about 100 EM runs with 100 iterations each, and it is unclear how to extend it for semi-supervised and supervised settings.[1]

In this article, we present a new approach to transliteration mining that is fully unsupervised like the system of Sajjad, Fraser, and Schmid (2011). It is based on a principled model which is both efficient and accurate and can be used in three different training settings—unsupervised, semi-supervised, and supervised learning. Our method directly learns character correspondences between two scripts from a noisy unlabeled list of word pairs which contains both transliterations and non-transliterations. When such a list is extracted from an aligned bilingual corpus, for instance, it contains,

1 There are other approaches to transliteration mining that exploit phonetic similarity between languages (Aransa, Schwenk, and Barrault 2012) and make use of temporal information available with the data (Tao et al. 2006). We do not discuss them here because they are out of the scope of this work.

apart from transliterations, also both translations and misalignments, which we will call **non-transliterations**.

Our statistical model interpolates a transliteration sub-model and a non-transliteration sub-model. The intuition behind using two sub-models is that the transliteration pairs and non-transliteration pairs, which make up the unlabeled training data, have rather different characteristics and need to be modeled separately. Transliteration word pairs show a strong dependency between source and target characters, whereas the characters of non-transliteration pairs are unrelated. Hence we use one sub-model for transliterations that jointly generate the source and target strings with a joint source channel model (Li, Min, and Jian 2004), and a second sub-model for non-transliterations that generate the two strings independently of each other using separate source and target character sequence models whose probabilities are multiplied.

The overall model is trained with the Expectation Maximization (EM) algorithm (Dempster, Laird, and Rubin 1977). Only the parameters of the transliteration model and the interpolation weight are learned during EM training, whereas the parameters of the non-transliteration model are kept fixed after initialization. At test time, a word pair is classified as a transliteration if the posterior probability of transliteration is higher than the posterior probability of non-transliteration.

For the *semi-supervised* system, we modify EM training by adding a new step, which we call the S-step. The S-step takes the probability estimates from one EM iteration on the unlabeled data and uses them as a backoff distribution in smoothing probabilities which were estimated from labeled data. The smoothed probabilities are then used in the next E-step. In this way, we constrain the parameters learned by EM to values that are close to those estimated from the labeled data.

In the *supervised* approach, we set the weight of the non-transliteration sub-model during EM training to zero, because all training word pairs are transliterations here. In test mode, the supervised mining model uses both sub-models and estimates a proper interpolation weight with EM on the test data.

We evaluate our system on the data sets available from the NEWS 2010 shared task on transliteration mining (Kumaran, Khapra, and Li 2010), which we call *NEWS10* herein. On three out of four language pairs, our unsupervised transliteration mining system performs better than all semi-supervised and supervised systems that participated in NEWS10. We also evaluate our unsupervised system on parallel corpora of English/Hindi and English/Arabic texts and show that it is able to effectively mine transliteration pairs from data with only 2% transliteration pairs.

The unigram version of the unsupervised and semi-supervised systems was published in Sajjad, Fraser, and Schmid (2012). In that paper, we proposed a supervised version of our transliteration mining system and also extended it to higher orders of character n-grams. Together with this article we also release data and source code as described below.

The contributions of this paper can be summarized as follows:

- We present a statistical model for unsupervised transliteration mining, which is very efficient and accurate. It models unlabeled data consisting of transliterations and non-transliterations.
- We show that our unsupervised system can easily be extended to both semi-supervised and supervised learning scenarios.
- We present a detailed analysis of our system using different types of corpora, with various learning strategies and with different n-gram orders.

We show that if labeled data are available, it is superior to building a semi-supervised system rather than an unsupervised system or a supervised system.

- We make our transliteration mining tool, which is capable of unsupervised, semi-supervised, and supervised learning, freely available to the research community at `http://alt.qcri.org/~hsajjad/software/transliteration_mining/`.
- We also provide the transliteration mining gold standard data that we created from English/Arabic and English/Hindi parallel corpora for use by other researchers.

2. Transliteration Mining Model

Our transliteration mining model is a mixture of a transliteration sub-model and a non-transliteration sub-model. The transliteration sub-model generates the source and target character sequences jointly and is able to model the dependencies between them. The non-transliteration model consists of two monolingual character sequence models that generate the source and target strings independently of each other.

The model structure is motivated as follows: A properly trained transliteration sub-model assigns most of the probability mass to transliteration pairs, whereas the non-transliteration sub-model evenly distributes the probability mass across all possible source and target word pairs. Hence a transliteration pair receives a high score from the transliteration sub-model and a low score from the non-transliteration sub-model, and vice versa for non-transliteration pairs. By comparing the scores of the two sub-models, we can classify word pairs. The interpolation weights of the two sub-models take the prior probabilities of transliteration and non-transliteration pairs into account.

The parameters of the two monolingual character sequence models of the non-transliteration sub-model are directly trained on the source and target part of the list of word pairs and are fixed afterwards. The parameters of the transliteration sub-model are uniformly initialized and then learned during EM training of the complete interpolated model. Why does this work? EM training is known to find a (local) maximum. The fixed non-transliteration sub-model assigns reasonable probabilities to any combination of source and target words (such as translations and misalignments) but fails to capture the dependencies between words and their transliterations. The only way for the EM training to increase the data likelihood is therefore a better modeling of transliteration pairs by means of the transliteration sub-model. After a couple of EM iterations, the transliteration sub-model is well adapted to transliterations and the interpolation weight models the relative frequencies of transliteration and non-transliteration pairs.

2.1 The Model

The *transliteration sub-model* creates a transliteration pair $(\mathbf{e}, \mathbf{f})$ consisting of a source word $\mathbf{e} = e_1 ... e_{|e|} = e_1^{|\mathbf{e}|}$ of length $|\mathbf{e}|$ and a target word $\mathbf{f} = f_1^{|\mathbf{f}|}$ of length $|\mathbf{f}|$ by generating a sequence of alignment units $\mathbf{a} = a_1^{|\mathbf{a}|}$ (later called *multigrams*).[2] Each multigram $a_i = (e_{x_{i-1}+1}^{x_i}, f_{y_{i-1}+1}^{y_i})$ comprises a substring $e_{x_{i-1}+1} ... e_{x_i}$ of the source word and a substring $f_{y_{i-1}+1} .. f_{y_i}$ of the target word $\mathbf{f}$ (with $x_0 = y_0 = 0$ and $x_{|\mathbf{a}|} = |\mathbf{e}|$ and $y_{|\mathbf{a}|} = |\mathbf{f}|$). The

2 The notation used in this section is partially borrowed from Bisani and Ney (2008).

Table 1
Two possible alignments of a word pair (*cef, ACDF*). The symbol ∅ represents the empty string.

Source word *cef*	∅	c	∅	e	f	∅	c	e	f
Target word *ACDF*	A	C	D	∅	F	A	C	D	F
Multigrams	∅-A	c-C	∅-D	e-∅	f-F	∅-A	c-C	e-D	f-F

substrings concatenated together form the source word and target word, respectively. In our experiments, the lengths of the substrings will be 0 or 1 (i.e., we have 0–1, 1–0, and 1–1 multigrams).

In general, there is more than one multigram sequence that generates a given transliteration pair. Table 1 shows two multigram sequences for the pair (*cef, ACDF*). We define the joint transliteration probability $p_1(\mathbf{e}, \mathbf{f})$ of a word pair as the sum of the probabilities of all multigram sequences:

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

where $Align(\mathbf{e}, \mathbf{f})$ returns all possible multigram sequences for the transliteration pair $(\mathbf{e}, \mathbf{f})$.

In a unigram model, the probability of a multigram sequence $\mathbf{a}$ is the product of the probabilities of the multigrams it contains:

$$p_1(\mathbf{a}) = p_1(a_1 a_2 ... a_{|\mathbf{a}|}) = \prod_{j=1}^{|\mathbf{a}|} p_1(a_j) \tag{2}$$

where $|\mathbf{a}|$ is the length of the sequence $\mathbf{a}$.

The *non-transliteration sub-model* generates source and target words that are unrelated. We model such pairs with two separate character unigram models (a source and a target model) whose probabilities are multiplied (Gale and Church 1993). Their parameters are learned from monolingual corpora and not updated during EM training. The non-transliteration sub-model is defined as follows:

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

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

The transliteration mining model is obtained by interpolating the transliteration model $p_1(\mathbf{e}, \mathbf{f})$ and the non-transliteration model $p_2(\mathbf{e}, \mathbf{f})$:

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

where λ is the prior probability of non-transliteration.

Interpolation with the non-transliteration model allows the transliteration model to concentrate on modeling transliterations during EM training. After EM training, transliteration word pairs are assigned a high probability by the transliteration

sub-model compared to the non-transliteration sub-model. Correspondingly, non-transliteration pairs are assigned a lower probability by the transliteration sub-model compared to the non-transliteration sub-model. This property is exploited to identify transliterations.

2.2 Model Estimation

In the following two subsections, we discuss the estimation of the parameters of the transliteration sub-model $p_1(\mathbf{e}, \mathbf{f})$ and the non-transliteration sub-model $p_2(\mathbf{e}, \mathbf{f})$.

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

For the transliteration model, we implemented a simplified form of the grapheme-to-phoneme converter g2p (Bisani and Ney 2008). g2p is able to learn general m-to-n character alignments between a source and a target word, although we restrict ourselves to 0–1, 1–1, and 1–0 alignments. Like Bisani and Ney (2008), we found in preliminary experiments that using more than one character on either or both sides of the multigram provides worse results.

Given a training corpus of N word pairs, the training data likelihood L can be calculated as the product of the probabilities of all training items. The EM algorithm is used to train the model. 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.

The expected count of a multigram a is computed by multiplying the posterior probability of each multigram sequence $\mathbf{a}$ with the frequency of a in $\mathbf{a}$ and summing these weighted frequencies over all alignments of all word pairs.

$$c(a) = \sum_{i=1}^{N} \sum_{\mathbf{a} \in Align(\mathbf{e}_i, \mathbf{f}_i)} p(\mathbf{a}|\mathbf{e}_i, \mathbf{f}_i) n_a(\mathbf{a}) \tag{5}$$

$n_a(\mathbf{a})$ is here the number of occurrences of the multigram a in the sequence $\mathbf{a}$. The posterior probability of $\mathbf{a}$ is given by:

$$p(\mathbf{a}|\mathbf{e}_i, \mathbf{f}_i) = \frac{(1-\lambda) p_1(\mathbf{a})}{p(\mathbf{e}_i, \mathbf{f}_i)}, \forall a \in Align(e_i, f_i) \tag{6}$$

where $p_1(\mathbf{a})$ is the probability of the alignment $\mathbf{a}$ according to the transliteration model (see Eq. 2). $1 - \lambda$ is the prior probability of transliteration, and $p(\mathbf{e}_i, \mathbf{f}_i)$ is defined in Equation (4).

We use relative frequency estimates to update the multigram probabilities of the unigram model. Besides the parameters of the transliteration model, we also need to reestimate the interpolation parameter λ. To this end, we sum up the posterior probabilities of non-transliteration over all training items and divide by the number of word pairs N to obtain a new estimate of λ.

In detail, this is done as follows: The posterior probability of non-transliteration $p_{ntr}(\mathbf{e}, \mathbf{f})$ is calculated by multiplying λ with the probability $p_2(\mathbf{e}, \mathbf{f})$ of the

non-transliteration model and normalizing the result by dividing it with the total probability of the word pair $p(\mathbf{e}, \mathbf{f})$.

$$p_{ntr}(\mathbf{e}, \mathbf{f}) = \frac{\lambda\, p_2(\mathbf{e}, \mathbf{f})}{p(\mathbf{e}, \mathbf{f})} \tag{7}$$

We calculate the expected count of non-transliterations by summing the posterior probabilities of non-transliteration over all word pairs:

$$c_{ntr} = \sum_{i=1}^{N} p_{ntr}(\mathbf{e}_i, \mathbf{f}_i) \tag{8}$$

λ is then re-estimated by dividing the expected count of non-transliterations by the number of word pairs N.

For the first EM iteration, the multigram probabilities are uniformly initialized with the inverse of the number of all possible multigrams that can be built from the source and target language characters. After the training, the prior probability of non-transliteration λ is an approximation of the fraction of non-transliteration pairs in the training data.

2.3 Implementation Details

We represent the character alignments of a word pair as a directed acyclic graph $G(N, E)$ with a set of nodes N and edges E. Each node of the graph corresponds to an index pair (i, j) and usually[3] has incoming edges from $(i-1, j-1)$ with label (e_i, f_j), from $(i-1, j)$ with label (e_i, ϵ), and from $(i, j-1)$ with label (ϵ, f_j), as well as outgoing edges to $(i+1, j+1)$, $(i+1, j)$, and $(i, j+1)$.

We implemented the Forward-Backward algorithm (Baum and Petrie 1966) to estimate the counts of the multigrams. The forward probability of a node sums the product of the forward probability of an incoming node and the probability of the multigram on the edge over all incoming nodes:

$$\alpha(s) = \sum_{r:(r,s)\in E} \alpha(r) p(a_{rs}) \tag{9}$$

r is the start node of the incoming edge to node s and the multigram a_{rs} is the label of the edge (r, s).

The backward probability $\beta(s)$ is computed in the opposite direction starting at the end node of the graph and proceeding to the first node. $\beta(|e|, |f|)$ and $\alpha(0, 0)$ are initially set to one.

3 Boundary nodes such as $(0, 0)$, $(0, 1)$, $(1, 0)$, $(|e|, |f|)$, and so forth, have fewer edges, of course.

Consider a node r connected to a node s via an edge labeled with the multigram a_{rs}. The expected count of a transition between r and s is calculated based on the forward and backward probabilities as follows:

$$\gamma'_{rs} = \frac{\alpha(r)p(a_{rs})\beta(s)}{\alpha(L)} \tag{10}$$

where $L = (|e|, |f|)$ is the final node of the graph whose forward probability is equal to the total probability $p_1(\mathbf{e}, \mathbf{f})$ of all multigram sequences.

In order to add transliteration information to every transition, we multiplied the expected count of a transition by the posterior probability of transliteration $1 - p_{ntr}(e,f)$, which in essence indicates how likely the string pair that contains the particular transition will be a transliteration pair. Note that non-transliterations are fixed during training.

$$\gamma_{rs} = \gamma'_{rs}(1 - p_{ntr}(e,f)) \tag{11}$$

$p_{ntr}(e,f)$ is defined in Equation (7).

The counts γ_{rs} are then summed for all multigram types a over all training pairs to obtain the frequencies $c(a)$. The new probability estimate of a multigram is calculated with relative frequencies.

3. Semi-Supervised Transliteration Mining

Our unsupervised transliteration mining system learns from unlabeled data only. It sometimes also extracts close transliterations (see Section 7.4 for details), which differ from true transliterations, for instance, by an inflectional ending added in one language. This has a negative impact on the precision of the system. Because of the lack of supervision, our unsupervised system also cannot adapt to different definitions of transliteration, which can vary from task to task.

Therefore we propose a semi-supervised extension of our unsupervised model that overcomes these shortcomings by using labeled data. The following subsections describe the model and the implementation details.

3.1 The Semi-Supervised Model

The semi-supervised system uses the same model as the unsupervised system, but is trained on both labeled and unlabeled data. The probability estimates learned on the labeled data are more accurate for frequent multigrams than the probabilities learned on the unlabeled data, but suffer from sparse data problems. Hence, we smooth the labeled data probability estimates with the unlabeled data probability estimates. The smoothed probability estimate $\hat{p}(a)$ is defined as follows:

$$\hat{p}(a) = \frac{c_s(a) + \eta_s p(a)}{N_s + \eta_s} \tag{12}$$

where $c_s(a)$ is the labeled data count of the multigram a, $p(a)$ is the unlabeled data probability estimate, $N_s = \sum_a c_s(a)$, and η_s is the number of different multigram types observed in the Viterbi alignment of the labeled data with the current model. The smoothing formula is motivated from Witten-Bell smoothing (Witten and Bell 1991).

3.2 Model Estimation

In the E-step on the labeled data, we set $\lambda = 0$ to turn off the non-transliteration model, which is not relevant here. This affects Equation (6), which defines the posterior probability of a multigram sequence **a**. For the unlabeled data, we initialize λ to 0.5 and recompute it as described in Section 2.2.

3.3 Implementation Details

We divide the training process of semi-supervised mining into two steps, which are illustrated in Figure 1. The goal of the first step is to create a reasonable initial alignment of the labeled data. This data set is small and might not be sufficient to learn good character alignments. Therefore we use the unlabeled data to help align it correctly. In contrast to the second step, we do not apply backoff smoothing in the first step,

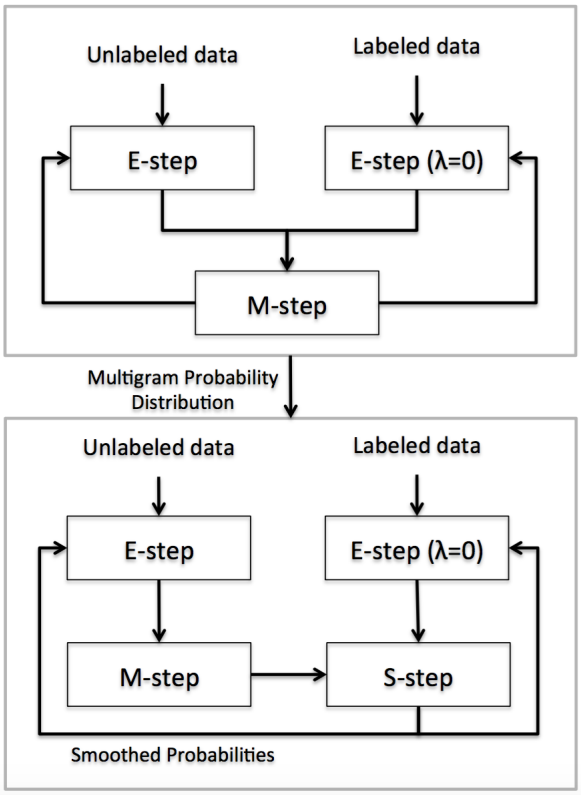


Figure 1
Semi-supervised training.

but simply sum the multigram counts obtained from the two data sets and compute relative frequency estimates. Apart from this, the first step is implemented in the same way as the second step.

The second step starts with the probability estimates from the first step and runs the E-step separately on labeled and unlabeled data. After the two E-steps, we estimate a new probability distribution from the counts obtained from the unlabeled data (M-step) and use it as a backoff distribution in computing smoothed probabilities from the labeled data counts (S-step). Figure 1 shows the complete procedure of semi-supervised training.

4. Supervised Transliteration Mining Model

For some language pairs, where a sufficient amount of labeled training data are available for transliteration mining, it is possibly better to use only labeled data because unlabeled data might add too much noise. This is the motivation for our supervised transliteration mining model.

4.1 Model Estimation

The supervised system uses the same model as described in Section 2, but the training data consist of transliteration pairs only, so the prior probability of non-transliteration λ is set to 0 during training. The parameters of the non-transliteration sub-model are trained on the source and target part of the training data as usual. The only model parameter that cannot be estimated on the labeled training data is the interpolation parameter λ. However, the test data consists of both transliterations and non-transliterations. We thus estimate this parameter on the test data as described in Section 2.2, while keeping the other parameters fixed.

Because of the sparsity of the labeled training data, some of the multigrams needed for the correct transliteration of the test words might not have been learned from the training data. We apply Witten-Bell smoothing with a uniform backoff distribution in order to assign a non-zero probability to them (see Section 6 for details).

4.2 Implementation Details

We use a similar implementation as described in Section 2.3. The training of the supervised mining system involves only labeled data, that is, transliterations. Therefore the posterior probability of transliteration in Equation (11) is 1 and we can directly use the values from Equation (10) as estimated multigram counts.

5. Higher Order Transliteration Mining Models

The unsupervised, semi-supervised, and supervised systems described in the previous sections were based on unigram models. We also experimented with higher order models that take preceding multigrams into account. In order to train a higher-order model, we first train a unigram model and compute the Viterbi alignments of the word pairs. The parameters of the higher order models are then directly estimated from the Viterbi alignments without further EM training.

In test mode, we compute the Viterbi alignment $\hat{\mathbf{a}}$ of each word pair $(\mathbf{e}, \mathbf{f})$, which is the most probable multigram sequence according to the n-gram model. It is defined as follows:

$$\hat{\mathbf{a}} = \arg \max_{\mathbf{a} \in Align(\mathbf{e},\mathbf{f})} \prod_{i=1}^{|\mathbf{a}|+1} p(a_i | a_{i-M+1}^{i-1}) \tag{13}$$

where M is the n-gram size and a_j with $j \leq 0$ and $j = |\mathbf{a}| + 1$ are boundary symbols.

The higher-order non-transliteration model probability is calculated according to Equation (3), but using higher order source and target language models, which are defined as $p_E(e) = \prod_{i=1}^{|e|} p_E(e_i | e_{i-M+1}^{i-1})$ and $p_F(f) = \prod_{i=1}^{|f|} p_F(f_i | f_{i-M+1}^{i-1})$. The posterior probability p_{ntr} of non-transliteration is computed based on the higher-order models according to Equation (7) and a word pair is classified as transliteration if $p_{ntr} < 0.5$ holds.

6. Smoothing to Deal with Unknowns in Testing

Our unsupervised and semi-supervised transliteration mining systems can be trained on one data set and tested on a different set. For the supervised system, training and test data are always different. In both scenarios, some of the multigrams needed to transliterate the test data might not have been learned from the training data. We apply Witten-Bell smoothing (Witten and Bell 1991) to assign a small probability to unknown characters and unknown multigrams. The smoothed probability of a multigram a is given by:

$$\hat{p}(a) = \frac{c(a) + \eta(.) p_{BO}(a)}{\sum_{a'} c(a') + \eta(.)} \tag{14}$$

$\eta(.)$ is the number of observed multigram types. We define $p_{BO}(a)$ as a uniform distribution over the set of all possible multigrams, i.e. $p_{BO}(a) = \frac{1}{(S+1)(T+1)}$ where S and T are the number of source and target language character types, respectively.

The parameters of the two monolingual n-gram models of the non-transliteration sub-model (see Equation (3)) are learned from the source and target words of the training data. Again, we use Witten-Bell smoothing to assign probabilities to unseen characters and character sequences. The model parameters are estimated analogous to Equation (14) with the following definition of the unigram probability:

$$\hat{p}_E(e) = \frac{c(e) + \eta(.) p_{EBO}(e)}{\sum_{e'} c(e') + \eta(.)} \tag{15}$$

$p_{EBO}(e)$ is obtained as follows: We assume that the number of character types is twice the number of character types $\eta(.)$ seen in the training data and define a uniform distribution $p_{EBO}(e) = \frac{1}{2\eta(.)}$. This smoothing formulation is equivalent to Add-λ smoothing with an additive constant of 0.5.

In the case of higher order models, some multigram sequences and character sequences might be unknown to the trained model. We also use Witten-Bell smoothing to estimate the probability of conditional multigram probabilities and character sequences.

The smoothing method slightly differs for the semi-supervised system, which is trained on both labeled and unlabeled data. The parameters of bigram models and higher are estimated on the labeled data, but the unigram model is smoothed with the unigram probability distribution estimated from unlabeled data, according to Equation (12).

7. Transliteration Mining Using the NEWS10 Data Set

We evaluate our system using the data set provided at the NEWS 2010 shared task on transliteration mining (Kumaran, Khapra, and Li 2010) (NEWS10). NEWS10 is a standard task on transliteration mining from Wikipedia InterLanguage Links, which are pairs of titles of Wikipedia pages that are on the same topic but written in different languages. Each data set contains training data, seed data, and reference data. The training data is a list of parallel phrases. The reference data is a small subset of the phrase pairs that have been annotated as transliterations or non-transliterations. The seed data is a list of 1,000 transliteration pairs provided to semi-supervised systems or supervised systems for initial training. We use the seed data only in our supervised and semi-supervised systems.

We evaluate on four language pairs English/Arabic, English/Hindi, English/Tamil, and English/Russian. We do not evaluate on the English/Chinese data because our extraction method was developed for languages with alphabetic script and probably needs to be adapted before it is applicable to logographic languages such as Chinese. One possible solution in the current set-up could be to convert Chinese to Pinyin and then apply transliteration mining. However, we did not try it under the scope of this work.

In unsupervised transliteration mining, our mining system achieves an improvement of up to 5% in F-measure over the heuristic-based unsupervised system of Sajjad et al. (2011). We also compare our unsupervised system with the semi-supervised and supervised systems presented at NEWS10 (Kumaran, Khapra, and Li 2010). Our unsupervised system outperforms all the semi-supervised and supervised systems that participated in NEWS10 on three language pairs.

7.1 Training Data

The NEWS10 training data consist of parallel phrases. In order to extract a candidate list of word pairs for training and mining, we take the cross-product by pairing every source word with all target words in the respective target phrase. We call it the **cross-product list** later on. Because of inconsistencies in the reference data, the list does not reach 100% recall. For example, the underscore is defined as a word boundary for English NEWS10 phrases. This assumption is not followed for certain phrases like "New_York" and "New_Mexico." There are 16, 9, 4, and 3 transliteration pairs missing out of 884, 858, 982, and 690 transliteration pairs in the cross-product list of English/Arabic, English/Russian, English/Hindi, and English/Tamil, respectively.

We preprocess the list and automatically remove numbers from the source and target language side because they are defined as non-transliterations (Kumaran, Khapra, and Li 2010). We also remove source language words that occur on the target language side and vice versa.

7.2 Experimental Set-up

Training. The *unsupervised system* is trained on the cross-product list only. The *semi-supervised system* is trained on the cross-product list and the seed data, and the *supervised system* is trained only on the seed data.

Parameters. The *multigram probabilities* are uniformly initialized with the inverse of the number of all possible multigrams of the source and target language characters. The *prior probability of non-transliteration* λ is initialized with 0.5.

Testing. In test mode, the trained model is applied to the test data. If the training data is identical to the test data, then the value of λ estimated during training is used at test time. If they are different, as in the case of the supervised system, we reestimate λ on the test data. Word pairs whose posterior probability of transliteration is above 0.5 are classified as transliterations.

7.3 Our Unsupervised System vs. State-Of-The-Art Unsupervised, Semi-Supervised, and Supervised Systems

Table 2 shows the result of our unsupervised transliteration mining system on the NEWS10 data set in comparison with the best unsupervised and (semi-)supervised systems presented at NEWS10 (S_{BEST}) and the best (semi-)supervised results reported overall on this data set (*GR*, *DBN*).

Our system performs consistently better than the heuristic-based system *SJD* for all experiments. On three language pairs, our unsupervised mining system performed better than all systems that participated in NEWS10. Its results are competitive with the best results reported on the NEWS10 data. On English/Hindi, our unsupervised system even outperforms all state-of-the-art supervised and semi-supervised systems. On the English/Russian data set, it faces problems with close transliterations, as further discussed in Section 7.6. Our semi-supervised extension of the system correctly classifies close transliterations as non-transliterations, as described in Section 7.4.

El-Kahki et al. (2011) (*GR*) achieved the best results on the English/Arabic, English/Tamil, and English/Russian data sets. For the English/Arabic task, they normalized the data using language dependent heuristics. They applied an Arabic word segmenter

Table 2
Comparison of our unsupervised system *OUR* with the state-of-the-art unsupervised, semi-supervised, and supervised systems, where S_{Best} is the best NEWS10 system, *SJD* is the unsupervised system of Sajjad et al. (2011), *GR* is the supervised system of Kahki et al. (2011), and *DBN* is the semi-supervised system of Nabende (2011). Noeman and Madkour (2010) have the best English/Arabic NEWS10 system. For all other language pairs, Jiampojamarn et al. (2010)'s system was the best in NEWS10.

	Unsupervised		**(Semi-) supervised systems**		
	OUR	*SJD*	S_{Best}	*GR*	*DBN*
English/Arabic	92.4	87.4	91.5	94.1	-
English/Hindi	95.7	92.2	92.4	93.2	95.5
English/Tamil	93.2	90.1	91.4	95.5	93.9
English/Russian	79.4	76.0	87.5	92.3	82.5

that uses language-dependent information. Arabic long vowels that have identical sound but are written differently were merged to one form. English characters were normalized by dropping accents. They also used a non-standard evaluation method (discussed in Section 7.6). Because of these heuristics, we consider their results not fully comparable with our results.

7.4 Comparison of Our Unigram Transliteration Mining Systems

In this section, we compare the unsupervised transliteration mining system with its semi-supervised and supervised variants.

Table 3 summarizes the results of the three systems on four language pairs. The unsupervised system achieves high recall with somewhat lower precision because it also extracts many close transliterations in particular for Russian (see Section 7.6 for details). The Russian data contain many pairs of words that differ only by their morphological endings. The unsupervised system learns to delete the endings with a high probability and incorrectly mines the word pairs.

On the non-Russian language pairs, the semi-supervised system achieves only a small gain in F-measure over the unsupervised mining system. This shows that the unlabeled training data already provides most of the transliteration information. The labeled data mostly helps the transliteration mining system to learn the exact definition of transliteration. This is most noticeable on the English/Russian data set, where the semi-supervised system achieves an almost 7% increase in precision with a 2.2% drop in recall compared with the unsupervised system. The F-measure gain is 3.7%. The increase in

Table 3
Results of our unsupervised, semi-supervised, and supervised transliteration mining systems trained on the cross-product list and using the unigram, bigram, and trigram models for transliteration and non-transliteration. The **bolded** values show the best precision, recall, and F-measure for each language pair.

	Unsupervised			**Semi-supervised**			**Supervised**		
	P	R	F	P	R	F	P	R	F
Unigram									
English/Arabic	89.2	**95.7**	**92.4**	**92.6**	92.2	**92.4**	84.9	95.1	89.7
English/Hindi	92.6	**99.0**	95.7	**95.5**	97.0	**96.3**	94.2	94.6	94.4
English/Tamil	88.3	**98.6**	93.2	**93.4**	95.8	**94.6**	90.4	95.8	93.0
English/Russian	67.1	**97.1**	79.4	**74.0**	94.9	**83.1**	70.0	95.3	80.7
Bigram									
English/Arabic	79.2	**96.4**	86.9	**93.4**	91.5	**92.5**	87.9	95.4	91.5
English/Hindi	81.6	**99.2**	89.5	**96.9**	95.3	**96.1**	95.3	94.1	94.7
English/Tamil	73.7	**98.8**	84.5	**95.2**	92.5	**93.8**	93.6	91.9	92.8
English/Russian	58.9	**98.0**	73.6	**77.7**	94.3	**85.2**	74.6	94.3	83.3
Trigram									
English/Arabic	58.4	**95.0**	72.4	**95.8**	83.4	**89.2**	82.0	94.6	87.9
English/Hindi	45.7	**97.5**	62.2	**97.8**	88.6	92.9	94.5	94.0	**94.2**
English/Tamil	33.5	**99.4**	50.1	**97.2**	84.3	90.3	92.6	90.7	**91.7**
English/Russian	38.1	**97.1**	54.7	**81.8**	88.0	**84.8**	72.7	95.3	82.2

precision shows that the labeled data helps the system in disambiguating transliteration pairs from close transliterations. For the English/Russian data set, we experimented with different sizes of the labeled data. Interestingly, using only 50 randomly selected labeled pairs results in an F-measure increase of 2 percentage points. This complements our mining model that learns transliteration from the unlabeled data and may need a small amount of labeled data only in special cases like the English/Russian data set.

The supervised system is only trained on the labeled data. It has higher precision than the unsupervised system except for English/Arabic, but lower recall, and for most language pairs the overall F-measure is below that of the unsupervised and semi-supervised systems. The reason for the low recall is that the labeled data consists of only 1,000 transliteration pairs, which is not enough for the model to learn good estimates for all parameters. Various multigrams that are needed to transliterate the test data have not been seen in the training data and the smoothed probability estimates are not informative enough. The re-estimation of the prior on the test data, which is different from the unsupervised and semi-supervised systems where the prior is re-estimated during EM training, could be another problem. The value of the prior has a direct effect on the posterior probability based on which the system extracts the transliteration pairs. In Section 7.7, we will show results from experiments where the threshold on the posterior probability of transliteration is varied. The supervised system achieved better F-measure at lower values of the posterior than 0.5, the value which works fine for most of the unsupervised and semi-supervised systems.

On the English/Russian data set, the supervised system achieves a 1.3% higher F-measure than the unsupervised system and about 3% better precision with a recall drop of 1.8%. The unsupervised system has problems with close transliterations as described before. The supervised system that is trained only on the labeled data is better able to correctly classify close transliterations.

These results can be summarized in the following way: The *semi-supervised system* has the best results for all four language pairs. The *unsupervised system* has the second-best results on three language pairs. It uses only unlabeled data for training, and thus could not differentiate between close transliterations and transliterations. The *supervised system* uses a small labeled data set for the training which is not sufficient to learn good estimates of all the multigrams. From the results of Table 3, we can conclude that if a small labeled data set is available, it is best to build a semi-supervised system. If no labeled data is available, an unsupervised system can be used instead, but might extract some spurious close transliterations.

7.5 Comparison of Our Higher-Order Transliteration Mining Systems

We now extend the unigram model to a bigram and trigram model as described in Section 5. Table 3 summarizes the results of our different higher-order systems on the four language pairs. For unsupervised systems, we see a consistent decline in F-measure for increasing n-gram order, which is caused by a large drop in precision. A possible explanation is that the higher-order unsupervised systems learn more noise from the noisy unlabeled data.

In the semi-supervised system, we see the opposite behavior: With growing n-gram order, precision increases and recall decreases. This could be explained by a stronger adaptation to the clean labeled data. In terms of F-measure, we only see a clear improvement over the unigram model for the English/Russian bigram model, where the context information seems to improve the classification of close transliterations.

The other results for the semi-supervised system are comparable or worse in terms of F-measure.

The supervised system shows similar tendencies as the semi-supervised system for the bigram model with an increase in precision and a drop in recall. The F-measure increases for English/Arabic and English/Russian and stays about the same for the other two languages pairs. A further increase of the *n*-gram order to trigrams leads to a general drop in all three measures except English/Russian recall.

We can conclude that a moderate increase of the *n*-gram size to bigrams helps the supervised system, hurts the unsupervised system, and benefits the semi-supervised system in the case of language pairs with many close transliterations.

We end this section with the following general recommendations:

- If no labeled data are available, it is best to use the unigram version of the unsupervised model for transliteration mining.
- If a small amount of labeled data are available, it is best to use a unigram or bigram semi-supervised system. Preference should be given to the bigram system when many close transliterations occur.
- The supervised system is not a good choice if the labeled data is as sparse as in our experiments.

7.6 Error Analysis

The errors made by our transliteration mining systems can be classified into the following categories.

Pronunciation Differences. Proper names may be pronounced differently in two languages. Sometimes, English short vowels are converted to long vowels in Hindi such as the English word "Lanthanum," which is pronounced "lʌnʈʰɑːnm" in Hindi. A similar case is the English/Hindi word pair "Donald/donɑːld." Sometimes two languages use different vowels to produce a similar sound like in the English word "January," which is pronounced as "ʤnʊriː" in Hindi. All these words only differ by one or two characters from an exact transliteration. According to the gold standard, they are non-transliterations but our unsupervised system classifies them as transliterations. The semi-supervised system is able to learn that they are non-transliterations. Table 4 shows a few examples of such word pairs.

Inconsistencies in the Gold Standard. There are a few word segmentation inconsistencies in the gold standard. The underscore "_" is defined as word boundary in the

Table 4
Word pairs with pronunciation differences.

English	Hindi	English	Hindi
Lanthanum	लाञथनम/lʌnʈʰnm	Sailendra	शैलेन्द्र/ʃɛːleːndr
January	जनवरी/dʒnʊriː	August	अगस्त/əgst

NEWS10 guidelines but this convention is not followed in the case of "New_York" and "New_Mexico." In the reference data, these phrases are included as single tokens with the "_" sign while all other phrases are word segmented on "_." We did not get these words in our training data as we tokenize all English words on "_."

Some Arabic nouns have an article "ال/ɑ:l" 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 "القران/ɑlqurɑ:n." Our semi-supervised mining system correctly classifies such cases as non-transliterations, but 24 of them are incorrectly annotated as transliterations in the gold standard. El-Kahki et al. (2011) preprocessed such Arabic words and separated "ال/ɑ:l" from the noun "القران/ɑlqurɑ:n" before mining. They report a match if the version of the Arabic word with "ال/ɑ:l" appears with the corresponding English word in the gold standard. Table 5 shows examples of word pairs that are wrongly annotated as transliterations in the gold standard.

Close Transliterations. Sometimes a word pair differs by only one or two ending characters from a true transliteration. Such word pairs are very common in the NEWS10 data. For example, in the English/Russian training data, the Russian nouns are often marked with case endings whereas their English counterparts lack such inflection. Because of the large number of such word pairs in the English/Russian data, our unsupervised transliteration 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. In the English/Hindi data set, such word pairs are mostly English words that are borrowed in Hindi like the word "calls," which is translated in Hindi as "cɑ:llæn." Table 6 shows some examples from the training data of English/Russian. All these pairs are mined by our systems as transliterations but marked as non-transliterations in the gold standard.

Table 5
Examples of word pairs that are wrongly annotated as transliterations in the gold standard.

English	Arabic	English	Arabic
Basrah	البصرة/ɑlbsʕrh	Nasr	النصر/ɑlnsʕr
Kuwait	الكويت/ɑlkwjt	Riyadh	الرياض/ɑlrjɑ:dʕ

Table 6
Close transliterations from the English/Russian corpus that are classified by our systems as transliteration pairs but labeled as non-transliterations in the gold standard.

English	Russian	English	Russian
Studio	Студия\ˈstudɪɟə	Geography	География\gʲɪɐˈgrafʲɪɟə
Estonia	Эстонии\əsˈtonʲɪ:	Monastery	Монастырь\mənɐsˈtʲɪr

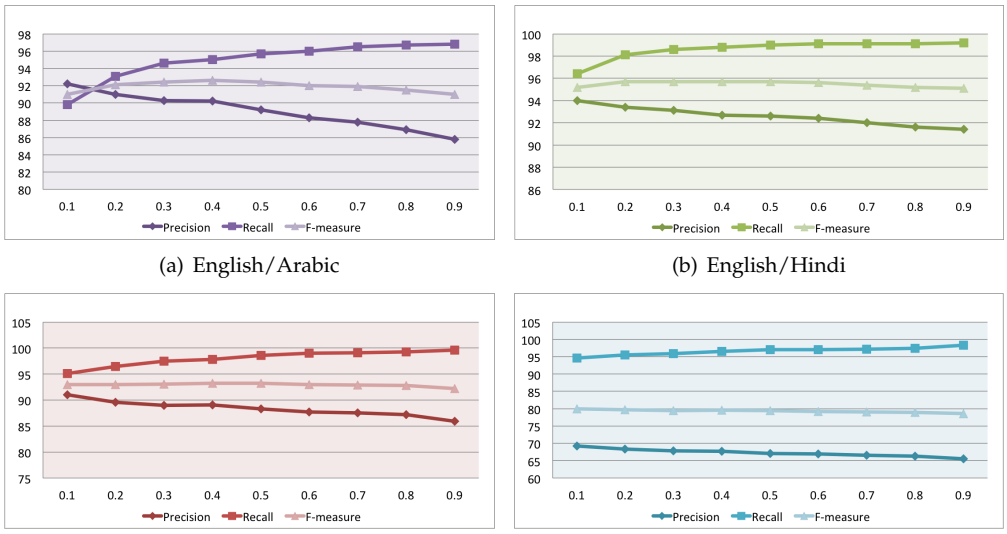
7.7 Variation of Parameters

Our transliteration mining system has two common parameters that need to be chosen by hand: the initial prior probability λ of non-transliteration and the classification threshold θ on the posterior probability of non-transliteration. The semi-supervised system has an additional smoothing parameter η_s, but its value is automatically calculated on the labeled data as described in Section 3.1. In all previous experiments, we used a value of 0.5 for both the prior λ and the threshold θ. Varying the value of λ, which is just used for initialization and then re-estimated during EM training, has little effect on the mined transliteration pairs and is therefore not considered further here. In this section, we examine the influence of the threshold parameter on the results.

Posterior Probability Threshold. Figure 2 summarizes the results of the unsupervised transliteration mining system obtained for different values of the threshold θ on the posterior probability of non-transliteration. For the unsupervised system, the value of $\theta = 0.5$ works fine for all language pairs and is either equal or close to the best F-measure the system is able to achieve.

Figure 3 shows the variation in the result of the semi-supervised system on different thresholds θ. The behavior is similar to the unsupervised system. However, the system achieves more balanced precision and recall and a higher F-measure than the unsupervised system.

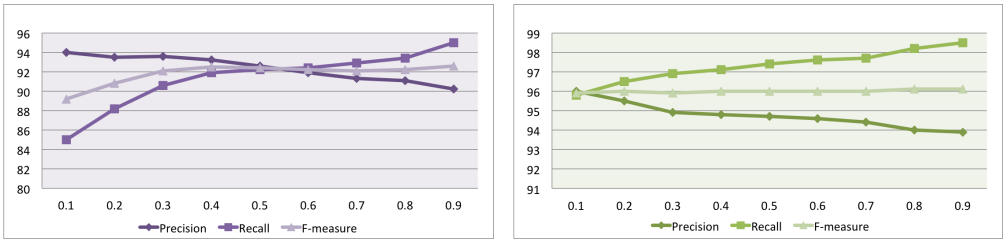
In contrast to the unsupervised and semi-supervised systems, the supervised transliteration mining system estimates the posterior probability of non-transliteration λ on the test data. Figure 4 shows the results of the supervised mining system using different thresholds θ on the posterior probability of non-transliteration. For all language pairs, the best F-measure is obtained at low thresholds around 0.05.



(a) English/Arabic (b) English/Hindi

(c) English/Tamil (d) English/Russian

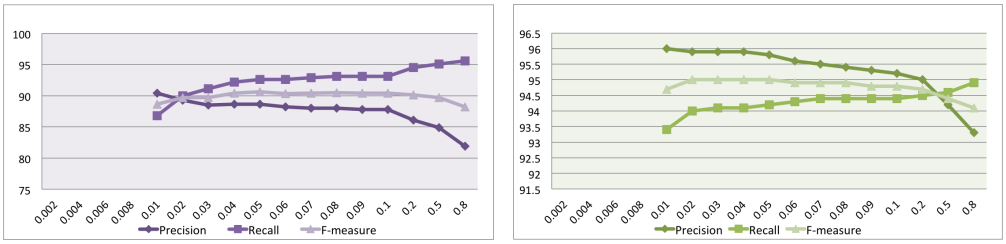
Figure 2
Effect of varying posterior probability threshold θ (x-axis) on the performance of the unigram unsupervised transliteration mining system.



(a) English/Arabic (b) English/Hindi

(c) English/Tamil (d) English/Russian

Figure 3
Effect of varying posterior probability threshold θ (*x*-axis) on the performance of the unigram semi-supervised transliteration mining system.



(a) English/Arabic (b) English/Hindi

(c) English/Tamil (d) English/Russian

Figure 4
Effect of varying posterior probability threshold θ (*x*-axis) on the performance of the unigram supervised transliteration mining system.

For all systems, keeping precision and recall balanced resulted in an F-measure value close to the best except for English/Russian where all variations of the model suffer from low precision.

Smoothing Parameter of the Semi-Supervised System. The smoothing technique used in the semi-supervised system is motivated from the Witten-Bell smoothing, which

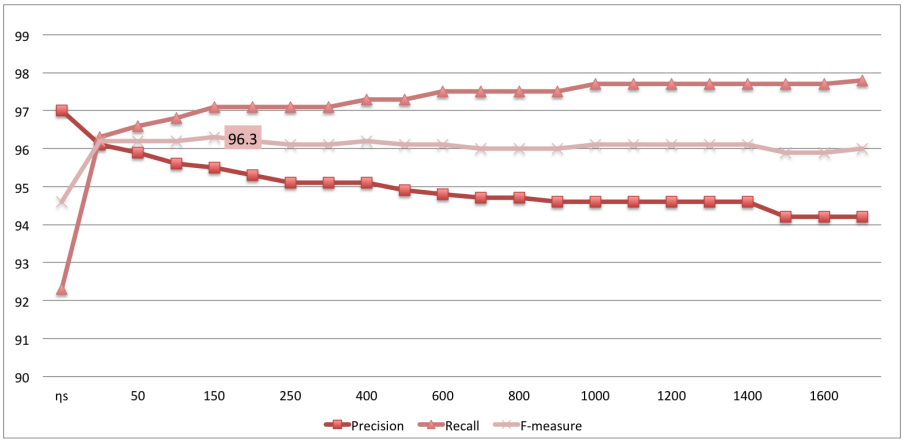


Figure 5
Results of the unigram semi-supervised mining system trained on the English/Hindi language pair using different values of η_s. 96.3 is the overall best value of F.

is normally applied to integer frequencies obtained by simple counting (see Equation (12)). We apply it to fractional counts obtained during EM training. We automatically choose the value of the smoothing parameter η_s as the number of different multigram types observed in the Viterbi alignment of the labeled data. This value works fine for all language pairs. Figures 5 and 6 show the variation of results for different values of η_s. The unigram system is trained on the English/Hindi and English/Russian cross-product list and the seed data. All results are calculated using $\theta = 0.5$. $\eta_s = 0$ means that the model assigns no weight to the unlabeled data and relies only on the smoothed labeled data probability distribution.

The system achieved an F-measure of 83.0 and 96.3 when using the automatically calculated value of η_s for English/Hindi and English/Russian, respectively. We can see

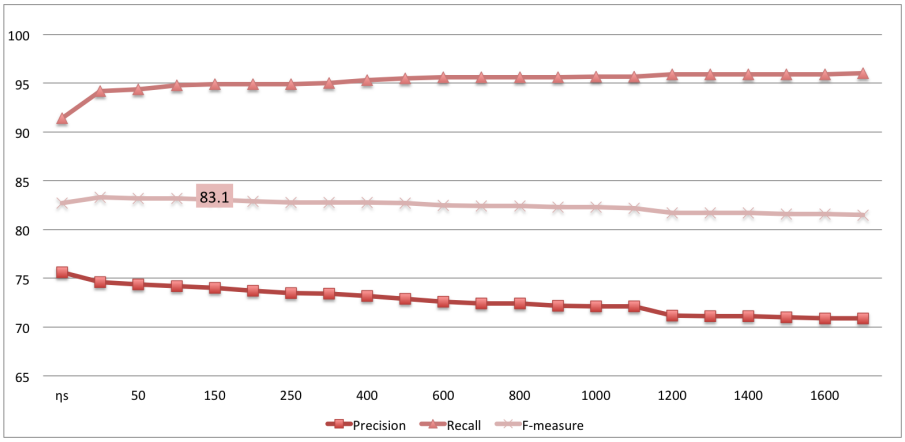


Figure 6
Results of the unigram semi-supervised mining system trained on the English/Russian language pair using different values of η_s. 83.1 is the overall best value of F.

that the automatically chosen values are close to the optimum highlighted in the figure. For all values $\eta_s > 0$, the English/Hindi system achieves a higher F-measure than the corresponding system, which relies only on the labeled data probabilities ($\eta_s = 0$). The same holds for the English/Russian system for values of η_s up to 600. As the weight of η_s increases, precision decreases and recall increases monotonically.

8. Transliteration Mining Using Parallel Corpora

The percentage of transliteration pairs in the NEWS10 data sets is much larger than in other parallel corpora. Therefore we also evaluated our transliteration mining system on new data sets extracted from English/Hindi and English/Arabic parallel corpora that have as few as 2% transliteration pairs. The English/Hindi corpus was published by the shared task on word alignment organized as part of the ACL 2005 Workshop on Building and Using Parallel Texts (WA05) (Martin, Mihalcea, and Pedersen 2005). For English/Arabic, we use 200,000 parallel sentences from the United Nations (UN) corpus (Eisele and Chen 2010) for the year 2000. We created gold standard annotations for these corpora for evaluation.

8.1 Training

We extract training data from the parallel corpora by aligning the parallel sentences using GIZA++ (Och and Ney 2003) in both directions, refine the alignments using the grow-diag-final-and heuristic (Koehn, Och, and Marcu 2003), and extract a *word-aligned list* from the 1-to-1 alignments.

We also build a cross-product list by taking all possible pairs of source and target words for each sentence pair. The cross-product list is huge and training a mining system on it would be too expensive computationally. So we train on the word-aligned list and use the cross-product list just for testing. Only the comparison of our unsupervised system and the heuristic-based system of Sajjad et al. (2011) is carried out on the word-aligned list. The cross-product list is noisier than the word-aligned list but misses fewer transliteration pairs. The English/Hindi cross-product list contains over two times more transliteration pairs (412 types) than the word-aligned list (180 types). The corresponding numbers for the English/Arabic cross-product list are not available because the English/Arabic gold standard was built on the word-aligned list. Table 7

Table 7
Statistics of the word-aligned list and the cross-product list of the English/Hindi and English/Arabic parallel corpus. The "Total" is the number of word pairs in the list. It is not equal to the sum of transliterations and non-transliterations in the list because the gold standard is only a subset of the training data.

	Translit	Non-translit	Total
English/Hindi$_{word-aligned}$	180	2,084	5,612
English/Hindi$_{cross-product}$	412	12,408	478,443
English/Arabic$_{word-aligned}$	288	6,639	178,342
English/Arabic$_{cross-product}$	288	6,639	26,782,146

shows the statistics of the word-aligned list and the cross-product list calculated using the gold standard of English/Hindi and English/Arabic.

8.2 Results

Our unsupervised system is only trained on the word-aligned list whereas our semi-supervised system is trained on the word-aligned list and the seed data provided by NEWS10 for English/Hindi and English/Arabic. The supervised system is trained on the seed data only. All systems are tested on the cross-product list. As always, we initialize multigrams with a uniform probability distribution in EM training and set the prior probability of non-transliteration initially to 0.5. At test time, the prior probability is reestimated on the test data because training and test data are different. A threshold of 0.5 on the posterior probability of non-transliteration is used for classification.

Deviating from the standard setting described above, we train and test our unsupervised unigram model-based system on the word-aligned list in order to compare it with the heuristic-based system, which cannot be run on the cross-product list for testing. Table 8 shows the results. On both languages, our system shows high recall of up to 100% with lower precision and achieves 0.6% and 1.8% higher F-measure than the heuristic-based system.

Table 9 shows the English/Hindi transliteration mining results of the unsupervised, semi-supervised, and supervised system trained on the word-aligned list and tested on the cross-product list. For all three systems, the unigram version performed best. Overall the unigram semi-supervised system achieved the best results with an F-measure of 85.6% and good precision as well as high recall. The unsupervised mining systems

Table 8
Transliteration mining results of the heuristic-based system *SJD* and the unsupervised unigram system *OUR* trained and tested on the word-aligned list of the English/Hindi and English/Arabic parallel corpus. TP, FN, TN, and FP represent true positive, false negative, true negative, and false positive, respectively.

	TP	FN	TN	FP	P	R	F
English/Hindi$_{SJD}$	170	10	2039	45	79.1	94.4	86.1
English/Hindi$_{OUR}$	176	4	2034	50	77.9	97.8	86.7
English/Arabic$_{SJD}$	197	91	6580	59	77.0	68.4	72.5
English/Arabic$_{OUR}$	288	0	6440	199	59.1	100.0	74.3

Table 9
Results of the unsupervised, semi-supervised, and supervised mining systems trained on the word-aligned list and tested on the cross-product list of the English/Hindi parallel corpus. The **bolded** values show the best precision, recall, and F-measure for the unigram, bigram, and trigram systems.

English/Hindi	Unsupervised			Semi-supervised			Supervised		
	P	R	F	P	R	F	P	R	F
Unigram	72.9	**93.9**	82.1	**79.8**	92.2	**85.6**	80.4	77.4	78.9
Bigram	4.2	**97.6**	8.1	**88.4**	81.3	**84.7**	79.9	76.0	77.9
Trigram	4.9	**97.6**	9.3	**87.2**	62.6	72.9	72.6	77.2	**74.8**

of higher order perform poorly because of very low precision. The higher-order semi-supervised systems show large drops in recall. The best F-measure achieved by the supervised system (78.9%) is much lower than the best F-measures obtained with the other systems. This is because of the small amount of labeled data used for the training of the supervised system in comparison to the huge unlabeled data.

The results on the English/Arabic parallel corpus are quite different (see Table 10). Here, the semi-supervised trigram system achieves the best F-measure. The unsupervised results are similar to those obtained for English/Hindi, but the precision of the unigram system is much lower, resulting in a low F-measure. Recall is excellent at 100%. The higher-order semi-supervised systems perform well here because the drop in recall is small.

In both experiments, the semi-supervised system better distinguishes between close transliterations and true transliterations than the unsupervised system. Table 11 shows a few word pairs from the English/Hindi experiment that were wrongly classified by the unigram unsupervised system and correctly classified by the unigram semi-supervised system. Although the unigram semi-supervised system is better than the unsupervised system, there are also a few close transliteration pairs that are wrongly classified by the unigram semi-supervised system. The bigram semi-supervised system exploits contextual information to correctly classify them. Table 12 shows a few word pairs from the English/Hindi experiment that are wrongly classified by the unigram semi-supervised system and correctly classified by the bigram semi-supervised system.

We observed that the error in the classification of close transliterations is an artifact of the data, which is extracted from a parallel corpus. Compared with the NEWS10 data, the parallel data contains a number of morphological variations of words. If a word occurs in several morphological forms, the miner learns to give high probability

Table 10
Results of the unsupervised, semi-supervised, and supervised mining systems trained on the word-aligned list and tested on the cross-product list of the English/Arabic parallel corpus. The **bolded** values show the best precision, recall, and F-measure for the unigram, bigram, and trigram systems.

English/Arabic	Unsupervised			Semi-supervised			Supervised		
	P	R	F	P	R	F	P	R	F
Unigram	41.1	**100.0**	58.3	51.4	99.3	67.7	**54.1**	97.9	**69.7**
Bigram	4.2	**100.0**	8.1	**61.3**	98.3	**75.5**	61.0	98.3	75.3
Trigram	4.2	**100.0**	8.1	**63.8**	97.2	**77.0**	44.9	98.6	61.7

Table 11
Examples of the English/Hindi close transliterations mined by the unigram unsupervised system and correctly classified as non-transliterations by the unigram semi-supervised system.

English	Hindi	English	Hindi
Also	एल/eːl	Schools	स्कूलों/skuːloːn
Buses	बसों/bsoːn	Trains	ट्रेनें/ʈrɛːnɛːn
Gas	जैसे/dʒɛːsen	‘your	योर/joːr

Table 12
Examples of the English/Hindi close transliterations mined by the unigram semi-supervised system and correctly classified as non-transliterations by the bigram semi-supervised system.

English	Hindi	English	Hindi
Appointments	अप्पाइंटमेंटों/əppɑːiːtm̃toːn	Homes	होम\ɦoːm
Brailed	ब्रैल/brɛːl	Miles	मील\miːl
Chemicals	कैमीकल\ kɛːmiːkl	Parked	पार्क\pɑːrk
Consumers'	कंज़यूमर्ज़\knzjuːmrz	Volunteering	वालंटरी\vɑːlntriː
Companies	कंपनियाँ\k̃pnɪjaːn		

to those character sequences that are common in all of its variations. This causes these close transliteration pairs to be identified as transliteration pairs.

We looked into the errors made by our bigram semi-supervised system. The mined transliteration pairs still contain close transliterations. These are arguably better than other classes of non-transliterations such as translations where source and target language words are unrelated at the character level.

Quantitative error analysis. We did a quantitative analysis of the errors made by the system for the English/Arabic language pair. We randomly selected 100 word pairs that were incorrectly mined as transliteration pairs and clustered them into the following error types: (1) *affix-based error*: the words differ by one or two characters at the start or end of the word; (2) *pronunciation error*: a borrowed word that sounds slightly different in the original language; (3) *punctuation errors*: one word has an additional punctuation symbol that makes the word pair a non-transliteration; (4) *gold standard error*: errors in the gold standard; (5) *worst errors*: word pairs that are far from being considered as transliteration pairs.

Table 13 shows the number of errors of each type. The affix-based and pronunciation errors are the top errors made by the system. Both of them plus punctuation errors come under the broad definition of close transliterations. These word pairs are helpful because they provide useful character-level transliteration information. Durrani et al. (2010) incorporated our unsupervised transliteration mining system into machine translation. They showed that for language pairs with fewer transliterations, the close transliterations help to build a stronger transliteration system.

Table 13
Types of errors made by the unsupervised transliteration mining system on the English/Arabic language pair. The numbers are based on randomly selected 100 word pairs that were wrongly classified by the mining system.

Error Type	Count	Error Type	Count	Error Type	Count
Affix-based Error	38	Pronunciation Error	22	Punctuation Error	10
Gold Standard Error	9	Worst Error	21	-	-

We observed that most of the word pairs in class 5 (worst errors) contain stop words. Because stop words are the most frequent words in a corpus, they occur with most of the target language words in the cross-product list. The unsupervised system starts learning wrong transliterations because of their high frequency. Durrani et al. (2010) preprocess the candidate list before mining the transliteration pairs and remove words pairs with the most common source and target words.

9. Applications

Our unsupervised transliteration mining system has been used in several NLP applications. Sajjad et al. (2013b) generated improved word alignment of the parallel training data by incorporating the transliteration mining module into GIZA++. Sajjad et al. (2013a), and Durrani et al. (2014) used transliteration mining to transliterate out-of-vocabulary words in a statistical machine translation system. They extracted transliteration pairs from the parallel corpus in an unsupervised fashion and trained a transliteration system on them. Kunchukuttan and Bhattacharyya (2015) showed the usefulness of unsupervised transliteration mining for the transliteration of Indian languages. Durrani and Koehn (2014) exploited the closeness between Urdu and Hindi using transliteration mining. They created synthetic Hindi-English parallel data by transliterating Urdu to Hindi. When they used the data for machine translation, it substantially improved translation quality.

10. Conclusions

We presented a statistical model for mining transliteration pairs from a list of candidate pairs in a fully unsupervised fashion. Our model consists of sub-models for transliterations and non-transliterations that are interpolated. The transliteration sub-model is an n-gram model over 1–1, 0–1, and 1–0 character pairs and the non-transliteration model is the product of two n-gram models over source and target characters, respectively. The statistical model is trained on the unlabeled data using the EM algorithm.

Our approach is completely unsupervised and works for all pairs of languages with an alphabetic script. Our best unsupervised mining system achieved F-measure values of up to 95.7% and outperformed all supervised and semi-supervised systems that participated in the NEWS10 shared task on three out of the four language pairs considered. On the English/Russian data set, accuracy was lower because many close transliterations (often cognates) were classified as transliterations. We extended our approach to semi-supervised and supervised mining that use labeled data for training. The supervised extension of our system performed poorly on most data sets because the small list of transliteration pairs available as labeled training data causes sparse data problems. The semi-supervised system resolved the limitations of our unsupervised and supervised systems. It achieved the best results and showed that labeled data helps the mining system to achieve high precision and that unlabeled data helps to avoid data sparseness problems.

Experiments with higher-order n-gram models showed mixed results. The unsupervised system did not profit from the wider context of higher-order models, but the bigram semi-supervised system generally showed high F-measure and retained a good balance between precision and recall.

From our experiments, we draw the following conclusions:

- If no labeled data is available for a language pair, it is best to build a unigram unsupervised transliteration mining system. The higher-order unsupervised systems tended to learn noise from the data and performed poorly.
- If some labeled data are available, it is always best to build a semi-supervised system. Higher-order semi-supervised systems learn contextual information, which helps to reach high precision. This is particularly useful on training data with a large number of close transliterations (such as cognates).

Acknowledgments

The authors wish to thank the anonymous reviewers. The research presented in this work has received funding from Deutsche Forschungsgemeinschaft grant Models of Morphosyntax for Statistical Machine Translation (Phase 2), Deutsche Forschungsgemeinschaft grant SFB 732, the European Research Council (ERC) under grant agreement no. 640550, the IST Program of the European Community under the PASCAL2 Network of Excellence, IST-2007-216886, and the Higher Education Commission of Pakistan. This publication only reflects the authors' views.

References

Aransa, Walid, Holger Schwenk and Loic Barrault. 2012. Semi-supervised transliteration mining from parallel and comparable corpora. In *Proceedings of the 9th International Workshop on Spoken Language Translation*, pages 185–192, Hong Kong.

Baum, Leonard E. and Ted Petrie. 1966. Statistical inference for probabilistic functions of finite state Markov chains. *Proceedings of the Annals of Mathematical Statistics*, 37(6):1554–1563.

Bisani, Maximilian and Hermann Ney. 2008. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451.

Darwish, Kareem. 2010, Transliteration mining with phonetic conflation and iterative training. In *Proceedings of the 2010 Named Entities Workshop*, 53–56, Uppsala.

Dempster, A. P., N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B (Methodological)*, 39(1):1–38.

Durrani, Nadir and Philipp Koehn. 2014. Improving machine translation via triangulation and transliteration. In *Proceedings of the 17th Annual Conference of the European Association for Machine Translation*, EAMT'14, pages 71–78, Dubrovnik.

Durrani, Nadir, Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2010. Hindi-to-Urdu machine translation through transliteration. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 465–474, Uppsala.

Durrani, Nadir, Hassan Sajjad, Hieu Hoang and Philipp Koehn. 2014. Integrating an unsupervised transliteration model into statistical machine translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 148–153, Gothenburg.

Eisele, Andreas and Yu Chen. 2010, MultiUN: A multilingual corpus from United Nation documents. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation*, pages 2868–2872, Valletta.

El-Kahki, Ali, 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*, pages 1384–1393, Edinburgh.

Gale, William A. and Kenneth W. Church. 1993. A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):177–184.

Huang, Fei. 2005. *Multilingual Named Entity Extraction and Translation from Text and Speech*. Ph.D. thesis, Language Technology Institute, Carnegie Mellon University.

Jiampojamarn, Sittichai, 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*, pages 39–47, Uppsala.

Koehn, Philipp, 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*, pages 48–54, Edmonton.

Kumaran, A., Mitesh M. Khapra, and Haizhou Li. 2010. Whitepaper of NEWS 2010 shared task on transliteration mining. In *Proceedings of the 2010 Named Entities Workshop*, pages 19–26, Uppsala.

Kunchukuttan, Anoop and Pushpak Bhattacharyya. 2015. Data representation methods and use of mined corpora for Indian language transliteration. In *Proceedings of the Fifth Named Entity Workshop*, pages 78–82, Beijing, Association for Computational Linguistics.

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

Martin, Joel, Rada Mihalcea, and Ted Pedersen. 2005. Word alignment for languages with scarce resources. In *ParaText '05: Proceedings of the Association for Computational Linguistics Workshop on Building and Using Parallel Texts*, pages 65–74, Morristown, NJ.

Nabende, Peter. 2010. Mining transliterations from Wikipedia using Pair HMMs. In *Proceedings of the 2010 Named Entities Workshop*, pages 76–80, Uppsala.

Noeman, Sara and Amgad Madkour. 2010. Language independent transliteration mining system using finite state automata framework. In *Proceedings of the 2010 Named Entities Workshop*, pages 112–115, Uppsala.

Och, Franz J. and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Sajjad, Hassan, Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. Comparing two techniques for learning transliteration models using a parallel corpus. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 129–137, Chiang Mai.

Sajjad, Hassan, 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*, pages 430–439, Portland, OR.

Sajjad, Hassan, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Conference of the Association for Computational Linguistics*, pages 469–477, Jeju Island.

Sajjad, Hassan, Francisco Guzmán, Preslav Nakov, Ahmed Abdelali, Kenton Murray, Fahad Al Obaidli, and Stephan Vogel. 2013a. QCRI at IWSLT 2013: Experiments in Arabic-English and English-Arabic spoken language translation. In *Proceedings of the 10th International Workshop on Spoken Language Technology (IWSLT-13)*, Heidelburg.

Sajjad, Hassan, Svetlana Smekalova, Nadir Durrani, Alexander Fraser, and Helmut Schmid. 2013b. QCRI-MES submission at WMT13: Using transliteration mining to improve statistical machine translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 219–224, Sofia.

Sherif, Tarek and Grzegorz Kondrak. 2007. Bootstrapping a stochastic transducer for Arabic-English transliteration extraction. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 864–871, Prague.

Tao, Tao, Su-Yoon Yoon, Andrew Fister, Richard Sproat, and ChengXiang Zhai. 2006. Unsupervised named entity transliteration using temporal and phonetic correlation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 250–257, Sydney.

Witten, Ian H. and Timothy C. Bell. 1991. The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression. *IEEE Transactions on Information Theory*, 37(4): 1085–1094.