Active Sample Selection for Named Entity Transliteration

Dan Goldwasser **Dan Roth**
Department of Computer Science
University of Illinois
Urbana, IL 61801
{goldwas1,danr}@uiuc.edu

Abstract

This paper introduces a new method for identifying named-entity (NE) transliterations within bilingual corpora. Current state-of-the-art approaches usually require annotated data and relevant linguistic knowledge which may not be available for all languages. We show how to effectively train an accurate transliteration classifier using very little data, obtained automatically. To perform this task, we introduce a new active sampling paradigm for guiding and adapting the sample selection process. We also investigate how to improve the classifier by identifying repeated patterns in the training data. We evaluated our approach using English, Russian and Hebrew corpora.

1 Introduction

This paper presents a new approach for constructing a discriminative transliteration model.
Our approach is fully automated and requires little knowledge of the source and target languages.
Named entity (NE) transliteration is the process of transcribing a NE from a source language to a target language based on phonetic similarity between the entities. Figure 1 provides examples of NE transliterations in English Russian and Hebrew.

Identifying transliteration pairs is an important component in many linguistic applications such as machine translation and information retrieval, which require identifying out-of-vocabulary words.
In our settings, we have access to source language NE and the ability to label the data upon request. We introduce a new active sampling paradigm that aims to guide the learner toward informative samples, allowing learning from a small number of representative examples. After the data is obtained it is analyzed to identify repeating patterns which can be used to focus the training process of the model.

English NE	Russian NE	Hebrew NE
Saint Petersburg	Санкт Петербург	סנקט פטרבורג

Figure 1: NE in English, Russian and Hebrew.

Previous works usually take a generative approach, (Knight and Graehl, 1997). Other approaches exploit similarities in aligned bilingual corpora; for example, (Tao et al., 2006) combine two unsupervised methods. (Klementiev and Roth, 2006) bootstrap with a classifier used interchangeably with an unsupervised temporal alignment method. Although these approaches alleviate the problem of obtaining annotated data, other resources are still required, such as a large aligned bilingual corpus.
The idea of selectively sampling training samples has been wildly discussed in machine learning theory (Seung et al., 1992) and has been applied successfully to several NLP applications (McCallum and Nigam, 1998). Unlike other approaches,our approach is based on minimizing the distance between the feature distribution of a comprehensive reference set and the sampled set.

2 Training a Transliteration Model

Our framework works in several stages, as summarized in Algorithm 1. First, a training set consisting

Proceedings of ACL-08: HLT, Short Papers (Companion Volume), pages 53–56,
Columbus, Ohio, USA, June 2008.

of NE transliteration pairs (w_s, w_t) is automatically generated using an active sample selection scheme. The sample selection process is guided by the Sufficient Spanning Features criterion (SSF) introduced in section 2.2, to identify informative samples in the source language.An oracle capable of pairing a NE in the source language with its counterpart in the target language is then used. Negative training samples are generated by reshuffling the terms in these pairs. Once the training data has been collected, the data is analyzed to identify repeating patterns in the data which are used to focus the training process by assigning weights to features corresponding to the observed patterns. Finally, a linear model is trained using a variation of the averaged perceptron (Freund and Schapire, 1998) algorithm. The remainder of this section provides details about these stages; the basic formulation of the transliteration model and the feature extraction scheme is described in section 2.1, in section 2.2 the selective sampling process is described and finally section 2.3 explains how learning is focused by using feature weights.

Input: Bilingual, comparable corpus ($\mathcal{S}$, $\mathcal{T}$), set of named entities NE_S from $\mathcal{S}$, Reference Corpus R_S, Transliteration Oracle O, Training Corpora D=D_S,D_T
Output: Transliteration model $\mathcal{M}$

Guiding the Sampling Process
repeat
  select a set C $\subseteq NE_S$ randomly
  $w_s = argmin_{w \in C} distance(R, D_S \cup \{w_s\})$
  $D = D \cup \{W_s, O(W_s)\}$
until *distance(R,*$D_S \cup \{W_s\}$*)* $\geq$ *distance(R,*D_S*)* ;
Determining Features Activation Strength
Define W:$f \rightarrow \Re$ s.t. foreach feature f=$\{f_s, f_t\}$
$W(f) = \frac{\sharp(f_s,f_t)}{\sharp(f_s)} \times \frac{\sharp(f_s,f_t)}{\sharp(f_t)}$
Use $\mathcal{D}$ to train $\mathcal{M}$;

Algorithm 1: Constructing a transliteration model.

2.1 Transliteration Model

Our transliteration model takes a discriminative approach; the classifier is presented with a word pair (w_s, w_t) , where w_s is a named entity and it is asked to determine whether w_t is a transliteration

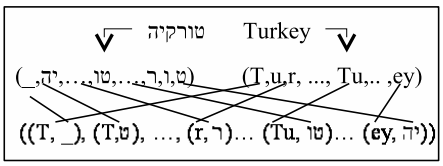

Figure 2: Features extraction process

of the NE in the target language. We use a linear classifier trained with a regularized perceptron update rule (Grove and Roth, 2001) as implemented in SNoW, (Roth, 1998). The classifier's confidence score is used for ranking of positively tagged transliteration candidates. Our initial feature extraction scheme follows the one presented in (Klementiev and Roth, 2006), in which the feature space consists of n-gram pairs from the two languages. Given a sample, each word is decomposed into a set of substrings of up to a given length (including the empty string). Features are generated by pairing substrings from the two sets whose relative positions in the original words differ by one or less places; first each word is decomposed into a set of substrings then substrings from the two sets are coupled to complete the pair representation. Figure 2 depicts this process.

2.2 Guiding the Sampling Process with SSF

The initial step in our framework is to generate a training set of transliteration pairs; this is done by pairing highly informative source language candidate NEs with target language counterparts. We developed a criterion for adding new samples, Sufficiently Spanning Features (SSF), which quantifies the sampled set ability to span the feature space. This is done by evaluating the L-1 distance between the frequency distributions of source language word fragments in the current sampled set and in a comprehensive set of source language NEs, serving as reference. We argue that since the features used for learning are n-gram features, once these two distributions are close enough, our examples space provides a good and concise characterization of all named entities we will ever need to consider. A special care should be given to choosing an appropriate reference; as a general guideline the reference set should be representative of the testing data. We collected a set R, consisting

of 50,000 NE by crawling through Wikipedia's articles and using an English NER system available at - *http://L2R.cs.uiuc.edu/ cogcomp*. The frequency distribution was generated over all character level bi-grams appearing in the text, as bi-grams best correlate with the way features are extracted. Given a reference text *R*, the n-grams distribution of *R* can be defined as follows -$D_R(ng_i) = \frac{\sharp ng_i}{\sum_j \sharp ng_j}$,where *ng* is an n-gram in R. Given a sample set *S*, we measure the L_1 distance between the distributions:
distance (R,S) = $\sum_{ng \in R} \mid D_R(ng) - D_S(ng) \mid$ Samples decreasing the distance between the distributions were added to the training data. Given a set C of candidates for annotation, a sample $w_s \in C$ was added to the training set, if -
$w_s = argmin_{w \in C} distance(R, D_S \cup \{w_s\})$.
A sample set is said to have SSF, if the distance remains constant as more samples are added.

2.2.1 Transliteration Oracle Implementation

The transliteration oracle is essentially a mapping between the named entities, i.e. given an NE in the source language it provides the matching NE in the target language. An automatic oracle was implemented by crawling through Wikipedia topic aligned document pairs. Given a pair of topic aligned documents in the two languages, the topic can be identified either by identifying the top ranking terms or by simply identifying the title of the documents. By choosing documents in Wikipedia's biography category we ensured that the topic of the documents is person NE.

2.3 Training the transliteration model

The feature extraction scheme we use generates features by coupling substrings from the two terms. Ideally, given a positive sample, it is desirable that paired substrings would encode phonetically similar or a distinctive context in which the two scripts correlate. Given enough positive samples, such features will appear with distinctive frequency. Taking this idea further, these features were recognized by measuring the co-occurrence frequency of substrings of up to two characters in both languages. Each feature *f=*(f_s, f_t) composed of two substrings taken from English and Hebrew words was associated with weight. $W(f) = \frac{\sharp(f_s,f_t)}{\sharp(f_s)} \times \frac{\sharp(f_s,f_t)}{\sharp(f_t)}$ where $\sharp(f_s, f_t)$ is the number of occurrences of that feature in the positive sample set, and $\sharp(f_L)$ is the number of occurrences of an individual substring, in any of the features extracted from positive samples in the training set. The result of this process is a weight table, in which, as we empirically tested, the highest ranking weights were assigned to features that preserve the phonetic correlation between the two languages. To improve the classifier's learning rate, the learning process is focused around these features. Given a sample, the learner is presented with a real-valued feature vector instead of a binary vector, in which each value indicates both that the feature is active and its activation strength - i.e. the weight assigned to it.

Data Set	Method	Rus	Heb
1	SSF	0.68	NA
1	KR'06	0.63	NA
2	SSF	0.71	0.52

Table 1: Results summary. The numbers are the proportion of NE recognized in the target language. Lines 1 and 2 compare the results of SSF directed approach with the baseline system on the first dataset. Line 3 summarizes the results on the second dataset.

3 Evaluation

We evaluated our approach in two settings; first, we compared our system to a baseline system described in (Klementiev and Roth, 2006). Given a bilingual corpus with the English NE annotated, the system had to discover the NE in target language text. We used the English-Russian news corpus used in the baseline system. NEs were grouped into equivalence classes, each containing different variations of the same NE. We randomly sampled 500 documents from the corpus. Transliteration pairs were mapped into 97 equivalence classes, identified by an expert. In a second experiment, different learning parameters such as selective sampling efficiency and feature weights were checked. 300 English-Russian and English-Hebrew NE pairs were used; negative samples were generated by coupling every English NE with all other target language NEs. Table 1 presents the key results of these experiments and compared with the baseline system.

Extraction method	Number of samples	Recall Top one	Recall Top two
Directed	200	0.68	0.74
Random	200	0.57	0.65
Random	400	0.63	0.71

Table 2: Comparison of correctly identified English-Russian transliteration pairs in news corpus. The model trained using selective sampling outperforms models trained using random sampling, even when trained with twice the data. The top one and top two results columns describe the proportion of correctly identified pairs ranked in the first and top two places, respectively.

Learning		Russian		Hebrew	
Training	Feature weights	Top one	Top five	Top one	Top five
+	+	0.71	0.89	0.52	0.88
-	+	0.63	0.82	0.33	0.59
+	-	0.64	0.79	0.37	0.68

Table 3: The proportion of correctly identified transliteration pairs with/out using weights and training. The top one and top five results columns describe the proportion of correctly identified pairs ranked in the first place and in any of the top five places, respectively. The results demonstrate that using feature weights improves performance for both target languages.

3.1 Using SSF directed sampling

Table 2 describes the effect of directed sampling in the English-Russian news corpora NE discovery task. Results show that models trained using selective sampling can outperform models trained with more than twice the amount of data.

3.2 Training using feature weights

Table 3 describes the effect training the model with weights.The training set consisted of 150 samples extracted using SSF directed sampling. Three variations were tested - training without feature weights, using the feature weights as the initial network weights without training and training with weights. The results clearly show that using weights for training improve the classifier's performance for both Russian and Hebrew. It can also be observed that in many cases the correct pair was ranked in any of the top five places.

4 Conclusions and future work

In this paper we presented a new approach for constructing a transliteration model automatically and efficiently by selectively extracting transliteration samples covering relevant parts of the feature space and focusing the learning process on these features. We show that our approach can outperform systems requiring supervision, manual intervention and a considerable amount of data. We propose a new measure for selective sample selection which can be used independently. We currently investigate applying it in other domains with potentially larger feature space than used in this work. Another aspect investigated is using our selective sampling for adapting the learning process for data originating from different sources; using the a reference set representative of the testing data, training samples, originating from a different source , can be biased towards the testing data.

5 Acknowledgments

Partly supported by NSF grant ITR IIS-0428472 and DARPA funding under the Bootstrap Learning Program.

References

Y. Freund and R. E. Schapire. 1998. Large margin classification using the perceptron algorithm. In *COLT*.

A. Grove and D. Roth. 2001. Linear concepts and hidden variables. *ML*, 42.

A. Klementiev and D. Roth. 2006. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *ACL*.

K. Knight and J. Graehl. 1997. Machine transliteration. In *EACL*.

D. K. McCallum and K. Nigam. 1998. Employing EM in pool-based active learning for text classification. In *ICML*.

D. Roth. 1998. Learning to resolve natural language ambiguities: A unified approach. In *AAAI*.

H. S. Seung, M. Opper, and H. Sompolinsky. 1992. Query by committee. In *COLT*.

T. Tao, S. Yoon, A. Fister, R. Sproat, and C. Zhai. 2006. Unsupervised named entity transliteration using temporal and phonetic correlation. In *EMNLP*.