1 Named Entity Transliteration and Discovery in Multilingual Corpora

Alexandre Klementiev
Dan Roth

Named Entity recognition (NER) is an important part of many natural language processing tasks. Current approaches often employ machine learning techniques and require supervised data. However, many languages lack such resources. This paper[1] *presents an (almost) unsupervised learning algorithm for automatic discovery of Named Entities (NEs) in a resource free language, given a bilingual corpora in which it is weakly temporally aligned with a resource rich language. NEs have similar time distributions across such corpora, and often some of the tokens in a multi-word NE are transliterated. We develop an algorithm that exploits both observations iteratively. The algorithm makes use of a new, frequency based, metric for time distributions and a resource free discriminative approach to transliteration. Seeded with a small number of transliteration pairs, our algorithm discovers multi-word NEs, and takes advantage of a dictionary (if one exists) to account for translated or partially translated NEs. We evaluate the algorithm on an English-Russian corpus, and show high level of NEs discovery in Russian.*

1.1 Introduction

Named Entity recognition has been getting much attention in NLP research in recent years, since it is seen as a significant component of higher level NLP tasks such as information distillation and question answering. Most modern approaches to

[1] This paper unifies and extends work from (Klementiev and Roth (2006a)) and (Klementiev and Roth (2006b)).

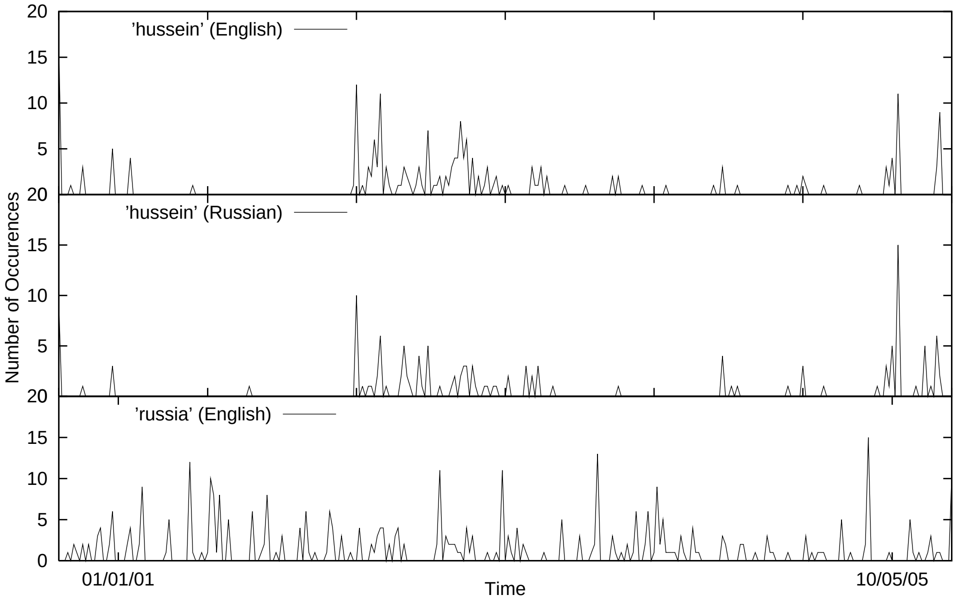

Figure 1.1 Temporal histograms for *Hussein* (top), its Russian transliteration (middle), and of the word *Russia* (bottom).

NER employ machine learning techniques, which require supervised training data. However, for many languages, these resources do not exist. Moreover, it is often difficult to find experts in these languages both for the expensive annotation effort and even for language specific clues. On the other hand, comparable multilingual data (such as multilingual news streams) are becoming increasingly available (see section 1.4). Unique properties of such corpora may allow us to transfer annotation across to resource poor domains, relieving the supervision bottleneck.

In this work, we make two independent observations about Named Entities encountered in such corpora, and use them to develop an algorithm that extracts pairs of NEs across languages. Specifically, given a bilingual corpora that is weakly temporally aligned, and a capability to annotate the text in one of the languages with NEs, our algorithm identifies the corresponding NEs in the second language text, and annotates them with the appropriate type, as in the source text.

The first observation is that NEs in one language in such corpora tend to co-occur with their counterparts in the other. E.g., Figure 1.1 shows a histogram of the number of occurrences of the word *Hussein* and its Russian transliteration in our bilingual news corpus spanning years 2001 through late 2005. One can see several common peaks in the two histograms, largest one being around the time of the beginning of the war in Iraq. The word *Russia*, on the other hand, has a distinctly different temporal signature. We can exploit such weak synchronicity of NEs across languages to associate them. In order to score a pair of entities across languages, we compute the similarity of their time distributions.

English NE	Russian NE
lilic	лилич
fletcher	флетчер
bradford	брэдфорд
isabel	изабель
hoffmann	гофман
kathmandu	катманду

Figure 1.2 Example English NEs and their transliterated Russian counterparts.

The second observation is that NEs often contain or are entirely made up of words that are phonetically transliterated or have a common etymological origin across languages (e.g. *parliament* in English and *парламент*, its Russian translation), and thus are phonetically similar. Figure 1.2 shows an example list of NEs and their possible Russian transliterations.

Approaches that attempt to use these two characteristics separately to identify NEs across languages would have significant shortcomings. Transliteration based approaches require a good model, typically handcrafted or trained on a clean set of transliteration pairs. On the other hand, time sequence similarity based approaches would incorrectly match words which happen to have similar time signatures (e.g., *Taliban* and *Afghanistan* in recent news).

We introduce an algorithm called *co-ranking*, which exploits these observations simultaneously to match NEs on one side of the bilingual corpus to their counterparts on the other.

We first train a transliteration model on single-word NEs. During training, for a given NE in one language, the current model chooses a list of top ranked transliteration candidates in another language. Discrete Fourier Transform (Arfken (1985)) based metric is then used to re-rank the list and choose the candidate best temporally aligned with the given NE. Finally, pairs of source language NEs and the top candidates from the re-ranked candidate lists are used for the next iteration of the transliteration model training.

Once the model is trained, NE discovery proceeds as follows. For a given NE, the transliteration model selects a candidate list for each constituent word. If a dictionary is available, each such candidate list is augmented with translations (if they exist). Translations will be the correct choice for some NE words (e.g. for *queen* in *Queen Victoria*), and transliterations for others (e.g. *Bush* in *Steven Bush*). We expect temporal sequence alignment to resolve many of such ambiguities. Temporal alignment score is used to re-rank translation/transliteration candidate lists for each constituent word. The top candidates from each re-ranked list are then merged into a possible target language NE. Finally, we verify that the candidate NE actually occurs in the target corpus.

A major challenge inherent in discovering transliterated NEs is the fact that a single entity may be represented by multiple transliteration strings. One reason is language morphology. For example, in Russian, depending on a case being used, the same noun may appear with various endings. Another reason is the lack of transliteration standards. Again, in Russian, several possible transliterations of an English entity may be acceptable, as long as they are phonetically similar to the source.

Thus, in order to rely on the time sequences we obtain, we need to be able to group variants of the same NE into an equivalence class, and collect their aggregate mention counts. We would then score time sequences of these equivalence classes. For instance, we would like to count the aggregate number of occurrences of {*Herzegovina, Hercegovina*} on the English side in order to map it accurately to the equivalence class of that NE's variants we may see on the Russian side of our corpus (e.g. {*Герцеговина, Герцеговину, Герцеговины, Герцеговиной*}). In the rest of the paper, whenever we refer to a Named Entity or an NE constituent word, we imply its equivalence class.

One of the objectives for this work was to use as little of the knowledge of both languages as possible. In order to effectively rely on the quality of time sequence scoring, we used a simple, knowledge poor approach to group NE variants for the languages of our corpus (see 1.3.1). Although we expect that better use of language specific knowledge would improve the results, it would defeat one of the goals of this work.

A demo of this work, as well as the software and the data used in the experiments are available at `http://L2R.cs.uiuc.edu/∼cogcomp/`.

1.2 Previous Work

There has been other work on discovering NEs automatically with minimal supervision. Both Cucerzan and Yarowsky (1999), and Collins and Singer (1999) present algorithms to obtain NEs from untagged corpora. However, they focus on the *classification* stage of already segmented entities, and make use of contextual and morphological clues that require knowledge of the language beyond the level we want to assume with respect to the target language.

The use of similarity of time distributions for information extraction, in general, and NE extraction, in particular, is not new. Hetland (2004) surveys recent methods for scoring time sequences for similarity. Shinyama and Sekine (2004) used the idea to discover NEs, but in a single language, English, across two news sources. Moreover, we use a different temporal distribution similarity function and show it to be better in section 1.4.3.

A large amount of previous work exists on transliteration models. Most are *generative* and consider the task of *producing* an appropriate transliteration for a given word, and thus require considerable knowledge of the languages. For example, AbdulJaleel and Larkey (2003); Jung et al. (2000) train English-Arabic and English-Korean generative transliteration models, respectively. Knight and Graehl (1997) build a generative model for backward transliteration from Japanese to English.

Sproat et al. (2006) produce transliterations by combining the scores of temporal and phonetic transliteration models, whereas we also propose a method to train a transliteration model with little supervision.

While generative models are often robust, they tend to make independence assumptions that do not hold in data. The discriminative learning framework argued for by Roth (1998, 1999) as an alternative to generative models is now used widely in NLP, even in the context of word alignment (Taskar et al. (2005); Moore (2005)). We make use of it here too, to learn a discriminative transliteration model that requires little knowledge of the target language.

1.3 *Co-Ranking*: An Algorithm for NE Discovery

In essence, the algorithm we present (Figure 1.3) uses temporal alignment as a supervision signal to iteratively train a transliteration model $\mathcal{M}$. On each iteration, for each NE in the source language corpus $\mathcal{S}$ it selects a list of top ranked transliteration candidates from the target language corpus $\mathcal{T}$ according to the current model (line 6). It then uses temporal alignment (with thresholding) to re-rank the list and select the best transliteration candidate for the next round of training (lines 8, and 10).

Similarly, in testing or discovery (Figure 1.4), candidate lists are collected (line 6) for each constituent word of each source NE using the trained model $\mathcal{M}$. Optionally, the lists $\mathcal{NE}^i_\mathcal{T}$ are augmented with the dictionary translations of the respective source word (line 7). The lists are then re-ranked without thresholding (line 9), and collected into a multi-word target NE candidate $\mathcal{E}_\mathcal{T}$. Finally, we discard $\mathcal{E}_\mathcal{T}$ which do not actually occur (in any order of the constituent words) in target corpus $\mathcal{T}$.

```
Algorithm Co-ranking [training]
Input:  Bilingual corpus ($\mathcal{S}$, $\mathcal{T}$), set of named entities $\mathcal{NE}_\mathcal{S}$ from $\mathcal{S}$, threshold $\theta$
Output:  Transliteration model $\mathcal{M}$
1.  Initialize $\mathcal{M}$.
2.  $\forall \mathcal{E} \in \mathcal{NE}_\mathcal{S}$, collect time distribution $\mathcal{Q}_{\mathcal{ES}}$.
3.  repeat
4.      $\mathcal{D} \leftarrow \emptyset$.
5.      for each $\mathcal{E}_\mathcal{S} \in \mathcal{NE}_\mathcal{S}$
6.          Use $\mathcal{M}$ to collect candidates $\mathcal{NE}_\mathcal{T} \in \mathcal{T}$ with high translit. scores.
7.          Collect time distribution $\mathcal{Q}_{\mathcal{ET}}$ for each candidate in $\mathcal{NE}_\mathcal{T}$.
8.          Select candidate $\mathcal{E}_\mathcal{T} \in \mathcal{NE}_\mathcal{T}$ with the best $\omega = score(\mathcal{Q}_{\mathcal{ES}}, \mathcal{Q}_{\mathcal{ET}})$.
9.          if $\omega > \theta$
10.             $\mathcal{D} \leftarrow \mathcal{D} \bigcup \{(\mathcal{E}_\mathcal{S}, \mathcal{E}_\mathcal{T})\}$.
11.     Use $\mathcal{D}$ to train $\mathcal{M}$
12. until Discovered training set $\mathcal{D}$ no longer changes between iterations.
13. return $\mathcal{M}$.
```

Figure 1.3 Iterative transliteration model training with single-word NEs.

```
Algorithm Co-ranking [testing]
Input: Bilingual corpus $(\mathcal{S}, \mathcal{T})$, set of named entities $\mathcal{NE}_\mathcal{S}$ from $\mathcal{S}$, transliteration model
    $\mathcal{M}$, dictionary dict (otional)
Output: Set of NE pairs $\mathcal{D}$ from $\mathcal{S}$ and $\mathcal{T}$
1.  $\mathcal{D} \leftarrow \emptyset$.
2.  for each $\mathcal{E}_\mathcal{S} \in \mathcal{NE}_\mathcal{S}$
3.          $\mathcal{E}_\mathcal{T} \leftarrow ()$.
4.          for each constituent word $\mathcal{E}^i_\mathcal{S}$ in $\mathcal{E}_\mathcal{S}$
5.                  Collect time distribution $\mathcal{Q}^i_{\mathcal{ES}}$ for $\mathcal{E}^i_\mathcal{S}$.
6.                  Use $\mathcal{M}$ to collect candidates $\mathcal{NE}^i_\mathcal{T} \in \mathcal{T}$ with high translit. scores.
7.                  (optional) $\mathcal{NE}^i_\mathcal{T} \leftarrow \mathcal{NE}^i_\mathcal{T} \bigcup dict(\mathcal{E}^i_\mathcal{S})$.
8.                  Collect time distribution $\mathcal{Q}^i_{\mathcal{ET}}$ for each candidate in $\mathcal{NE}^i_\mathcal{T}$.
9.                  Select candidate $\mathcal{E}^i_\mathcal{T} \in \mathcal{NE}^i_\mathcal{T}$ with the best $\omega = score(\mathcal{Q}^i_{\mathcal{ES}}, \mathcal{Q}^i_{\mathcal{ET}})$.
10.                 $\mathcal{E}_\mathcal{T} \leftarrow \mathcal{E}_\mathcal{T} + \mathcal{E}^i_\mathcal{T}$.
11.         if $Occurs(\mathcal{E}_\mathcal{T})$
12.                 $\mathcal{D} \leftarrow \mathcal{D} \bigcup \{(\mathcal{E}_\mathcal{S}, \mathcal{E}_\mathcal{T})\}$.
13. return $\mathcal{D}$.
```

Figure 1.4 Testing phase.

1.3.1 Time sequence generation and matching

In order to generate a time sequence for a given word, we sort the set of (time-stamped) documents of our corpus into a sequence of equally sized temporal bins. We then count the number of occurrences of the word in each bin, and normalize the sequence.

We use a method called the F-index (Hetland (2004)) to implement the *score* similarity function (Figure 1.3, line 8, and Figure 1.4, line 9). We first run a Discrete Fourier Transform on a time sequence to extract its Fourier expansion coefficients. The score of a pair of time sequences is then computed as a Euclidean distance between their expansion coefficient vectors.

As we mentioned in the introduction, an NE may map to more than one transliteration in another language. Identification of the entity's equivalence class of transliterations is important for accurately obtaining its time sequence.

In order to keep to our objective of requiring as little language knowledge as possible, we took a rather simplistic approach for both languages of our corpus. For Russian, two words were considered variants of the same NE if they share a prefix of size five or longer. Each unique word had its own equivalence class for the English side of the corpus, although, in principal, ideas such as in (Li et al. (2004)) could be incorporated. A cumulative distribution was then collected for such equivalence classes.

1.3.2 Transliteration model

Unlike most of the previous work considering *generative* transliteration models, we take the *discriminative* approach. Indeed, we do not need generate transliterations for unseen Named Entities. Instead, we aim to match NEs in the source language to their counterparts present in the target language side of our corpus in order to transfer annotation.

We train a linear model to decide whether a word $\mathcal{E}_{\mathcal{T}} \in \mathcal{T}$ is a transliteration of an NE $\mathcal{E}_{\mathcal{S}} \in \mathcal{S}$. The words in the pair are partitioned into a set of substrings $s_{\mathcal{S}}$ and $s_{\mathcal{T}}$ up to a particular length (including the empty string $\_$). Couplings of the substrings $(s_{\mathcal{S}}, s_{\mathcal{T}})$ from both sets produce features we use for training. Note that couplings with the empty string represent insertions/omissions.

Consider the following example: $(\mathcal{E}_{\mathcal{S}}, \mathcal{E}_{\mathcal{T}})$ = (powell, пауэлл). We build a feature vector from this example in the following manner:

1. We split both words into all possible substrings of up to size two:
   - $\mathcal{E}_{\mathcal{S}} \rightarrow \{\_, p, o, w, e, l, l, po, ow, we, el, ll\}$
   - $\mathcal{E}_{\mathcal{T}} \rightarrow \{\_,$ *п, а, у, э, л, л, па, ау, уэ, эл, лл*$\}$
2. We then build a feature vector by coupling substrings from the two sets:
   - $((p, \_), (p,$ *а*$), ...(w,$ *уэ*$), ...(el,$ *эл*$), ...(ll,$ *лл*$))$

We use the observation that transliteration tends to preserve phonetic sequence to limit the number of couplings. For example, we can disallow the coupling of substrings whose starting positions are too far apart: thus, we might not consider a pairing (po, ue) in the above example. In our experiments, we paired substrings if their positions in their respective words differed by -1, 0, or 1.

We use the perceptron (Rosenblatt (1958)) algorithm to train the model. The model activation provides the score we use to select best transliterations on line 6. Our version of perceptron takes variable number of features in its examples; each example is a subset of all features seen so far that are active in the input. As the iterative algorithm observes more data, it discovers and makes use of more features. This model is called the *infinite attribute model* (Blum (1992)) and it follows the perceptron version of SNoW (Carlson et al. (1999)).

Positive examples used for iterative training are pairs of NEs and their best temporally aligned transliteration candidates. Alignment score thresholding is used to implement the tradeoff between the quality and the number of the positive examples selected for the next round. Negative examples are English non-NEs paired with random Russian words.

1.4 Experimental Study

We ran experiments using a bilingual comparable English-Russian news corpus we built by crawling a Russian news web site (`www.lenta.ru`). The site provides loose translations of (and pointers to) the original English texts. We collected pairs of articles spanning from 1/1/2001 through 10/05/2005. The corpus consists of 2,327 documents, with 0-8 documents per day[2]. The English side was tagged with a publicly available NER system based on the SNoW learning architecture (Carlson et al. (1999)), that is available on the same site. This set of English NEs was hand-pruned to remove incorrectly classified words to obtain 978 single word NEs.

Temporal distributions were collected with bin size of one day, as described in 1.3.1. In order to reduce running time, some limited pre-processing was done on the Russian side. All equivalence classes, whose temporal distributions were close to uniform (i.e. words with a similar likelihood of occurrence throughout the corpus) were deemed common and not considered as NE candidates. Unique words were thus grouped into 14,781 equivalence classes.

Unless mentioned otherwise, the transliteration model was initialized with a set of 20 pairs of English NEs and their Russian transliterations. Negative examples here and during the rest of the training were pairs of non-NE English and Russian words selected uniformly randomly from the respective corpora.

As the transliteration model improves throughout training, new examples and thus new features are discovered. All but top 3000 features from positive and 3000 from negative examples were pruned based on the number of their occurrences so far. Features remaining at the end of training were used for NE discovery.

Insertions/omissions features (see 1.3.2) were not used in the experiments as they provided no tangible benefit for the languages of our corpus.

In each iteration, we used the current transliteration model to find a list of 30 best transliteration equivalence classes for each NE. We then computed time sequence similarity score between NE and each class from its list to find the one with the best matching time sequence. If its similarity score surpassed a set threshold, it was added to the list of positive examples for the next round of training. Positive examples were constructed by pairing an NE with the common stem of its transliteration equivalence class. We used the same number of positive and negative examples.

We used the Mueller English-Russian dictionary to obtain translations in our multi-word NE experiments. Lists of transliteration candidates were augmented with up to 10 dictionary translation.

For evaluation, random 727 of the total of 978 NEs were matched to correct transliterations by a language expert (partly due to the fact that some of the English NEs were not mentioned in the Russian side of the corpus). Accuracy was computed as the percentage of NEs correctly identified by the algorithm. Note that although multiple correct Russian transliterations are possible for a given English NE, the evaluation set included only a single one (due to the prohibitive amount of labor required of the language expert otherwise). Thus, evaluation results tend to be conservative.

[2] The corpus, code and demo are available at `http://L2R.cs.uiuc.edu/∼cogcomp/`.

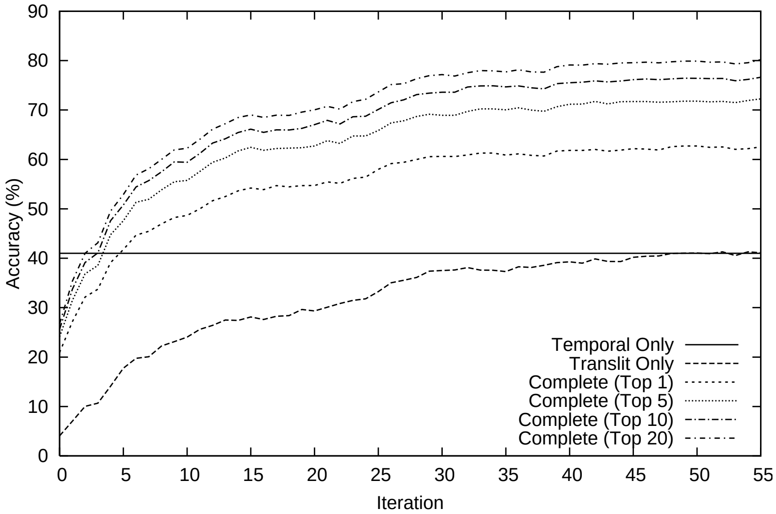

Figure 1.5 Proportion of correct NEs in top N discovered candidates vs. training iteration (averaged over 5 runs initialized with different random sets of 20 examples). The complete algorithm outperforms both transliteration model and temporal sequence matching when used on their own.

In the multi-word NE experiment, 177 random multi-word (2 or more) NEs and their transliterations/translations discovered by the algorithm were verified by a language expert. Again, phrases which were incorrectly tagged as NEs by the source language NE tagger were discarded.

1.4.1 NE discovery

1.4.1.1 Single-word NEs

Figure 1.5 shows the proportion of correctly discovered NE transliteration equivalence classes throughout the training stage. The figure also shows the accuracy if transliterations are selected according to the current transliteration model (top scoring candidate) and temporal sequence matching alone.

The complete algorithm experiments included counting if the correct transliteration appeared as the top scoring candidate (*Top 1*), was present in top five (*Top 5*), top ten (*Top 10*), or top twenty (*Top 20*) candidates chosen by the algorithm.

Both the transliteration model and the temporal alignment alone achieve the accuracy of about 41%. The combined algorithm achieves about 63%, showing a significant improvement over either of the two methods alone. Moreover, the correct NEs appear among the top 5 candidates 72% of the time, among top 10 - 77%, and among top 20 - 80%.

Feature	Num. in neg.	Num. in pos.	Percent in pos.
(a, а)	1432	6820	82.65
(n, н)	1182	5517	82.36
(r, р)	1011	5278	83.92
(gh, г)	5	137	96.48
(hm, м)	0	100	100
(tz, ц)	0	78	100
(j, х)	3	71	95.95
(j, дж)	0	198	100
(ic, ич)	11	403	97.34
(an, ан)	22	1365	98.41

Figure 1.6 A sample of the features discovered by the algorithm during training.

1.4.1.2 Discovered Features

Figure 1.6 lists a few interesting features discovered by the algorithm during training. As expected, single letter pairs which have similar pronunciation in both languages are highly indicative of a transliteration. English two-letter sequences *gh* and *hm* correspond to a single letter sequences in Russian, since *h* is often silent. Letter *j* is pronounced differently in names of Hispanic origin and is thus mapped to two distinct letter sequences in Russian. Some features are particularly useful for the specific training corpus. For example, the news corpus often refers to Serbian surnames ending in *ic*.

1.4.1.3 Intuition

In order to understand what happens to the transliteration model as the training proceeds, let us consider the following example. Figure 1.7 shows parts of candidate transliteration lists[3] for NE *forsyth* for two iterations of the algorithm. The weak transliteration model selects the correct transliteration (italicized) as the 24th best transliteration in the first iteration. Time sequence scoring function chooses it to be one of the training examples for the next round of training of the model. By the eighth iteration, the model has improved to select it as a best transliteration.

Not all correct transliterations make it to the top of the candidates list (transliteration model by itself is never as accurate as the complete algorithm on Figure 1.5). That is not required, however, as the model only needs to be good enough to place the correct transliteration anywhere in the candidate list.

Not surprisingly, some of the top transliteration candidates start sounding like the NE itself, as training progresses. On Figure 1.7, candidates for *forsyth* on iteration 7 include *fross* and *fossett*.

[3] Each candidate is represented by an equivalence class: a common prefix and a set of endings found in text.

	Iteration 0		Iteration 7
1	скоре {-е, -й, -йшего, -йший}	1	*форсайт {-а, -, -у}*
2	оформ {-лено, -ил, ...}	2	оформ {-лено, -ил, -ить, ...}
3	кокрэйн {-а, -}	3	проры {-вом, -ва, -ли, ...}
4	флоре {-нс, -нц, -, -нции}	4	фросс
	•	5	фоссет {-т, -та, -ту, -а, -у}
	•		•
24	*форсайт {-а, -, -у}*		•
	•		•

Figure 1.7 Lists of Russian transliteration candidates for *forsyth* for two iterations of the algorithm. As transliteration model improves, the correct transliteration moves up the list.

1.4.1.4 Multi-word NEs

Once the transliteration model was trained, we ran the algorithm to discover multi-word NEs, augmenting candidate sets of dictionary words with their translations as described in Section 1.3. Of all multi-word Named Entity pairs discovered by the algorithm, about 68% were matched correctly. The discovered Russian NEs included entirely transliterated, partially translated, and entirely translated NEs. Some of them are shown on Figure 1.8.

1.4.2 Initial example set size

We ran a series of experiments to see how the size of the initial training set affects the accuracy of the model as training progresses (Figure 1.9). Although the performance of the early iterations is significantly affected by the size of the initial training example set, the algorithm quickly improves its performance. As we decrease the size from 80 to 20 and then to 5, the accuracy of the first iteration drops by over 15% and 10% respectively. However, in about 50 iterations all three perform similarly.

English NE	**Russian NE equivalence class**
carla del ponte	карла{-, -йл} дель понте
marc dutroux	марк дютру
pangbourne	пангбурн
supreme council	верхо{-вный, ...} совет{...}
congolese	конго{-, -лезской}
north carolina	север{...} карол{-ина, ...}
junichiro koizumi	дзюнитиро коидзуми
rehman	реман{-, -а}

Figure 1.8 Example of correct transliterations discovered by the algorithm.

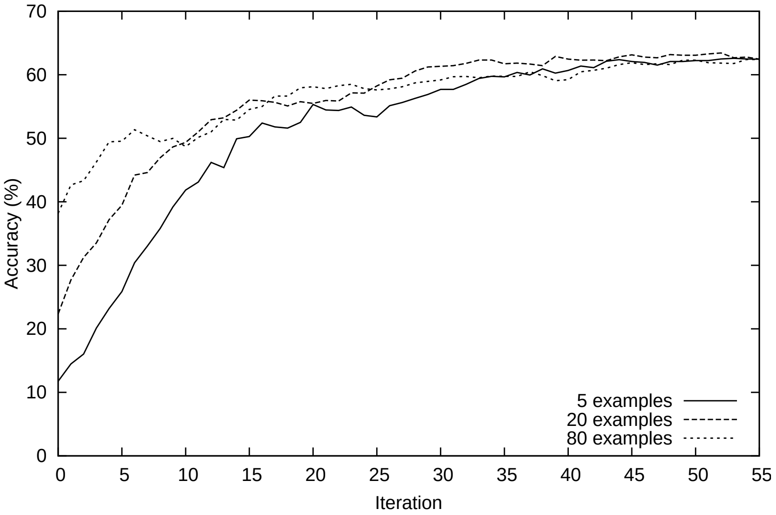

Figure 1.9 Proportion of correctly discovered NE pairs vs. the initial example set size (averaged over 3 runs each). Decreasing the number of examples does not have an impact on the performance of the later iterations.

The few examples in the initial training set produce features corresponding to substring pairs characteristic for English-Russian transliterations. Model trained on these (few) examples chooses other transliterations containing the same substring pairs. In turn, the chosen positive examples contain other characteristic substring pairs, which will be used by the model (via the *infinite attribute domain*, Blum (1992)) to select more positive examples on the next round, and so on. The smaller the initial training set, the longer it takes to discover the characteristic features, and the longer it takes for the algorithm to converge.

One would also expect the size of the training set necessary for the algorithm to improve to depend on the level of temporal alignment of the two sides of the corpus. Indeed, the weaker the temporal supervision the more we need to endow the model so that it can select cleaner candidates in the early iterations.

1.4.3 Comparison of time sequence scoring functions

We compared the DFT-based time sequence similarity scoring function we use in this paper to the commonly used *cosine* (Salton and McGill (1986)) and *Pearson*'s correlation measures in order to assess its performance and robustness to misalignment between two sides of the corpus.

We perturbed the Russian side of the corpus in the following way. Articles from each day were randomly moved (with uniform probability) within a k-day window. We ran single word NE temporal sequence matching alone on the perturbed corpora using each of the three measures (Figure 1.10).

	***k*=1**	***k*=3**	***k*=5**
Cosine	41.3	5.8	1.7
Pearson	41.1	5.8	1.7
DFT	41.0	12.4	4.8

Figure 1.10 Proportion of correctly discovered NEs vs. corpus misalignment (k) for each of the three measures. DFT based measure provides significant advantages over commonly used metrics for weakly aligned corpora.

	***w*=1**	***w*=2**	***w*=3**
Cosine	5.8	13.5	18.4
Pearson	5.8	13.5	18.2
DFT	12.4	20.6	27.9

Figure 1.11 Proportion of correctly discovered NEs vs. sliding window size (w) for each of the three measures.

Some accuracy drop due to misalignment could be accommodated for by using a larger temporal bin for collecting occurrence counts. We tried various (sliding) window size w for a perturbed corpus with $k = 3$ (Figure 1.11).

DFT metric outperforms the other measures significantly in most cases. NEs tend to have distributions with few pronounced peaks. If two such distributions are not well aligned, we expect both Pearson and cosine measures to produce low scores, whereas the DFT metric should catch their similarities in the frequency domain.

1.5 Conclusions

We have proposed a novel algorithm for cross lingual multi-word NE discovery in a bilingual weakly temporally aligned corpus. We have demonstrated that using two independent sources of information (transliteration and temporal similarity) together to guide NE extraction gives better performance than using either of them alone (see Figure 1.5).

The algorithm requires almost no supervision, or linguistic knowledge. Indeed, we used a very small bootstrapping training set and made a simple assumption in order to group morphological variants of the same word into equivalence classes in Russian.

We also developed a linear discriminative transliteration model, and presented a method to automatically generate features. For time sequence matching, we used a scoring metric novel in this domain. We provided experimental evidence that this metric outperforms other scoring metrics traditionally used.

1.6 Future Work

The algorithm can be naturally extended to comparable corpora of more than two languages. Pair-wise time sequence scoring and transliteration models should give better confidence in NE matches.

The ultimate goal of this work is to automatically tag NEs so that they can be used for training of an NER system for a new language. To this end, we would like to compare the performance of an NER system trained on a corpus tagged using this approach to one trained on a hand-tagged corpus.

1.7 Acknowledgments

We thank Richard Sproat, ChengXiang Zhai, and Kevin Small for their useful feedback during this work, and the anonymous referees for their helpful comments. This research is supported by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program and a DOI grant under the Reflex program.

References

Nasreen AbdulJaleel and Leah S. Larkey. Statistical transliteration for english-arabic cross language information retrieval. In *Proceedings of CIKM*, pages 139–146, New York, NY, USA, 2003.

George Arfken. *Mathematical Methods for Physicists.* Academic Press, 1985.

Avrim Blum. Learning boolean functions in an infinite attribute space. *Machine Learning*, 9(4):373–386, 1992.

A. Carlson, C. Cumby, J. Rosen, and D. Roth. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May 1999. URL `http://l2r.cs.uiuc.edu/~danr/Papers/CCRR99.pdf`.

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 1999.

Silviu Cucerzan and David Yarowsky. Language independent named entity recognition combining morphological and contextual evidence. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, 1999.

Magnus Lie Hetland. *Data Mining in Time Series Databases*, chapter A Survey of Recent Methods for Efficient Retrieval of Similar Time Sequences. World Scientific, 2004.

Sung Young Jung, SungLim Hong, and Eunok Paek. An english to korean transliteration model of extended markov window. In *Proc. the International Conference on Computational Linguistics (COLING)*, pages 383–389, 2000.

Alexandre Klementiev and Dan Roth. Named entity transliteration and discovery from multilingual comparable corpora. In *Proc. of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2006a.

Alexandre Klementiev and Dan Roth. Weakly supervised named entity transliteration and discovery from multilingual comparable corpora. In *Proc. of the Annual Meeting of the Association of Computational Linguistics*, 2006b.

Kevin Knight and Jonathan Graehl. Machine transliteration. In *Proc. of the Meeting of the European Association of Computational Linguistics*, pages 128–135, 1997.

Xin Li, Paul Morie, and Dan Roth. Identification and tracing of ambiguous names: Discriminative and generative approaches. In *Proceedings of the National*

Conference on Artificial Intelligence (AAAI), pages 419–424, 2004.

Robert C. Moore. A discriminative framework for bilingual word alignment. In *Proc. of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 81–88, 2005.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 1958.

Dan Roth. Learning to resolve natural language ambiguities: A unified approach. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 806–813, 1998.

Dan Roth. Learning in natural language. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 898–904, 1999.

Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill, Inc., New York, NY, USA, 1986.

Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *Proc. the International Conference on Computational Linguistics (COLING)*, pages 848–853, 2004.

Richard Sproat, Tao Tao, and ChengXiang Zhai. Named entity transliteration with comparable corpora. In *Proc. of the Annual Meeting of the Association of Computational Linguistics*, pages 73–80, 2006.

Ben Taskar, Simon Lacoste-Julien, and Michael Jordan. Structured prediction via the extragradient method. In *The Conference on Advances in Neural Information Processing Systems (NIPS).* MIT Press, 2005.

Index