Named Entity Transliteration and Discovery from Multilingual Comparable Corpora

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

Abstract

Named Entity recognition (NER) is an important part of many natural language processing tasks. Most current approaches employ machine learning techniques and require supervised data. However, many languages lack such resources. This paper presents an algorithm to automatically discover Named Entities (NEs) in a resource free language, given a bilingual corpora in which it is weakly temporally aligned with a resource rich language. We observe that NEs have similar time distributions across such corpora, and that they are often transliterated, and develop an algorithm that exploits both iteratively. The algorithm makes use of a new, frequency based, metric for time distributions and a resource free discriminative approach to transliteration. We evaluate the algorithm on an English-Russian corpus, and show high level of NEs discovery in Russian.

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, and an enabling technology for better information access. Most successful approaches to 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 increasingly available (see section 4).

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 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 as a way to associate them. In order to score a pair of entities across languages, we compute the similarity of their time distributions.

The second observation is that NEs are often transliterated or have a common etymological origin across languages, and thus are phonetically similar. Figure 2 shows an example list of NEs and their pos-

Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pages 82–88,
New York, June 2006.

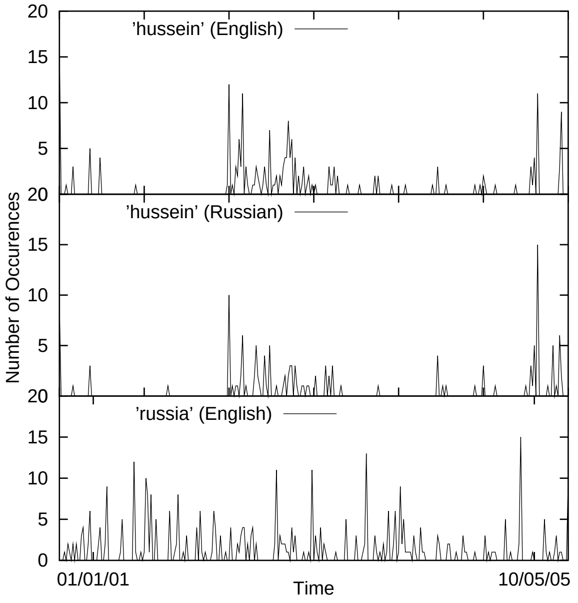

Figure 1: Temporal histograms for *Hussein* (top), its Russian transliteration (middle), and of the word *Russia* (bottom).

sible 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 we call *co-ranking* which exploits these observations simultaneously to match NEs on one side of the bilingual corpus to their counterparts on the other. We use a Discrete Fourier Transform (Arfken, 1985) based metric for computing similarity of time distributions, and we score NEs similarity with a linear transliteration model. For a given NE in one language, the transliteration model chooses a top ranked list of candidates in another language. Time sequence scoring is then used to re-rank the candidates and choose the one best temporally aligned with the NE. That is, we attempt to choose a candidate which is both a good transliteration (according to the current model) and is well aligned with the NE. Finally, pairs of NEs and the best candidates are used to iteratively train the transliteration model.

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

Figure 2: Example English NEs and their transliterated Russian counterparts.

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*}).

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 Russian.

In the rest of the paper, whenever we refer to a Named Entity, we imply an NE equivalence class. Note that although we expect that better use of language specific knowledge would improve the results, it would defeat one of the goals of this work.

2 Previous Work

There has been other work to automatically discover NE 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.

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.

While generative models are often robust, they tend to make independence assumptions that do not hold in data. The discriminative learning framework argued for in (Roth, 1998; Roth, 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.

3 *Co-ranking*: An Algorithm for NE Discovery

In essence, the algorithm we present uses temporal alignment as a supervision signal to iteratively train a discriminative transliteration model, which can be viewed as a distance metric between and English NE and a potential transliteration. On each iteration, it selects a set of transliteration candidates for each NE according to the current model (line 6). It then uses temporal alignment (with thresholding) to select the best transliteration candidate for the next round of training (lines 8, and 9).

Once the training is complete, lines 4 through 10 are executed without thresholding for each NE in $\mathcal{S}$ to discover its counterpart in $\mathcal{T}$.

3.1 Time Sequence Generation and Matching

In order to generate time sequence for a word, we divide the corpus into a sequence of temporal bins, and count the number of occurrences of the word in each bin. We then normalize the sequence.

We use a method called the F-index (Hetland, 2004) to implement the $score$ similarity function on line 8 of the algorithm. 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 Euclidian distance between their expansion coefficient vectors.

.

```
   Input: Bilingual, comparable 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 $\mathcal{E}_\mathcal{S} \in \mathcal{NE}_\mathcal{S}$ do
6          Use $\mathcal{M}$ to collect a set of candidates $\mathcal{NE}_\mathcal{T} \in \mathcal{T}$
           with high transliteration scores;
7          $\forall \mathcal{E} \in \mathcal{NE}_\mathcal{T}$ collect time distribution $\mathcal{Q}_{\mathcal{ET}}$;
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 $\omega$ exceeds $\theta$, add tuple $(\mathcal{E}_\mathcal{S}, \mathcal{E}_\mathcal{T})$ to $\mathcal{D}$ ;
10     end
11     Use $\mathcal{D}$ to train $\mathcal{M}$;
12 until D stops changing between iterations ;
```

Algorithm 1: Co-ranking: an algorithm for iterative cross lingual NE discovery.

3.1.1 Equivalence Classes

As we mentioned earlier, an NE in one language may map to multiple morphological variants and transliterations in another. Identification of the entity's equivalence class of transliterations is important for obtaining its accurate time sequence.

In order to keep to our objective of requiring as little language knowledge as possible, we took a rather simplistic approach to take into account morphological ambiguities of NEs in Russian. Two words were considered variants of the same NE if they share a prefix of size five or longer. At this point, our algorithm takes a simplistic approach also for the English side of the corpus – each unique word had its own equivalence class although, in principle, we can incorporate works such as (Li et al., 2004) into the algorithm. A cumulative distribution was then collected for such equivalence classes.

3.2 Transliteration Model

Unlike most of the previous work to transliteration, that consider *generative* transliteration models, we take a *discriminative* approach. 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, pauel). We build a feature vector from this example in the following manner:

- First, we split both words into all possible substrings of up to size two:

  $\mathcal{E}_\mathcal{S} \rightarrow \{\_, p, o, w, e, l, l, po, ow, we, el, ll\}$

  $\mathcal{E}_\mathcal{T} \rightarrow \{\_, p, a, u, e, l, pa, au, ue, el\}$

- We build a feature vector by coupling substrings from the two sets:

  $((p, \_), (p, a), ...(w, au), ...(el, el), ...(ll, el))$

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 examples with a variable number of features; each example is a set 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 in SNoW (Roth, 1998).

Positive examples used for iterative training are pairs of NEs and their best temporally aligned (thresholded) transliteration candidates. Negative examples are English non-NEs paired with random Russian words.

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 12/24/2004. The corpus consists of 2,022 documents with 0-8 documents per day. The corpus is available on our web page at `http://L2R.cs.uiuc.edu/~cogcomp/`. The English side was tagged with a publicly available NER system based on the SNoW learning architecture (Roth, 1998), that is available at the same site. This set of English NEs was hand-pruned to remove incorrectly classified words to obtain 978 single word NEs.

In order to reduce running time, some limited preprocessing was done on the Russian side. All 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 grouped into 15,594 equivalence classes, and 1,605 of those classes were discarded using this method.

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

Unless mentioned otherwise, the transliteration model was initialized with a subset of 254 pairs of NEs and their transliteration equivalence classes. Negative examples here and during the rest of the training were pairs of randomly selected non-NE English and Russian words.

In each iteration, we used the current transliter-

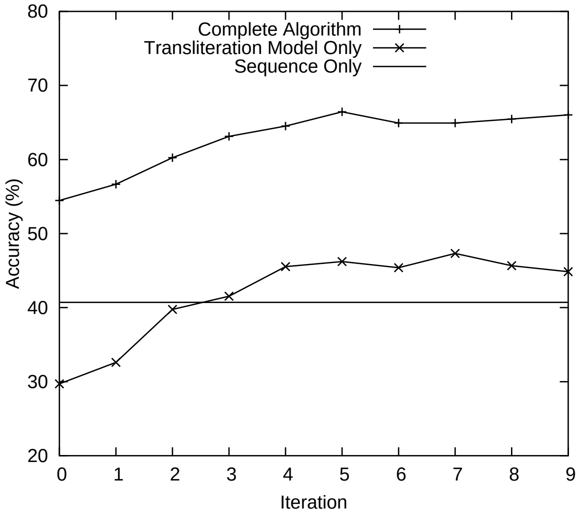

Figure 3: Proportion of correctly discovered NE pairs vs. iteration. Complete algorithm outperforms both transliteration model and temporal sequence matching when used on their own.

ation 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 each English NE with each of the transliterations from the best equivalence class that surpasses the threshold. We used the same number of positive and negative examples.

For evaluation, random 727 of the total of 978 NE pairs matched by the algorithm were selected and checked by a language expert. Accuracy was computed as the percentage of those NEs correctly discovered by the algorithm.

4.1 NE Discovery

Figure 3 shows the proportion of correctly discovered NE transliteration equivalence classes throughout the run of the algorithm. The figure also shows the accuracy if transliterations are selected according to the current transliteration model (top scoring candidate) and sequence matching alone. The transliteration model alone achieves an accuracy of about 47%, while the time sequence alone gets about 41%. The combined algorithm achieves about 66%, giving a significant improvement.

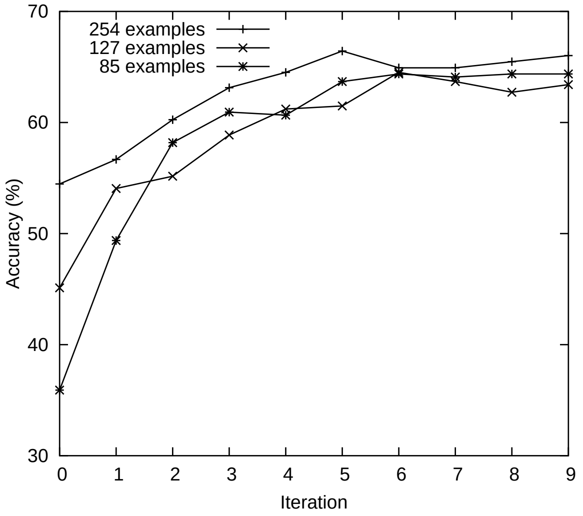

Figure 5: Proportion of the correctly discovered NE pairs for various initial example set sizes. Decreasing the size does not have a significant effect of the performance on later iterations.

In order to understand what happens to the transliteration model as the algorithm proceeds, let us consider the following example. Figure 4 shows parts of transliteration lists 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 3). 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 4, candidates for *forsyth* on iteration 7 include *fross* and *fossett*.

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

Figure 4: Transliteration lists for *forsyth* for two iterations of the algorithm ranked by the current transliteration model. As the model improves, the correct transliteration moves up the list.

4.2 Rate of Improvement vs. 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 5). 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 254, to 127, to 85 examples, the accuracy of the first iteration drops by roughly 10% each time. However, starting at the 6th iteration, the three are with 3% of one another.

These numbers suggest that we only need a few initial positive examples to bootstrap the transliteration model. The intuition is the following: 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 these same substring pairs. In turn, the chosen positive examples contain other characteristic substring pairs, which will be used by the model to select more positive examples on the next round, and so on.

5 Conclusions

We have proposed a novel algorithm for cross lingual 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 3).

We 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. As supported by our own experiments, this method outperforms other scoring metrics traditionally used (such as *cosine* (Salton and McGill, 1986)) when corpora are not well temporally aligned.

English NE	Russian NE equiv. class
lincoln	линкольн {-а, -, -шир}
oregon	орегон {-ского, -е}
niznansky	низнански
uruguay	ругва {-йское, -йцы, -я, -й}
rosing	розингом
gruban	грубан {-ом, -}
meiwes	майвес {-а, -у}
rosetta	розеттского
ecuador	эквадор {-а, -}
laxman	лакшман
friedrich	фридрих {-, -а}
chad	чада

Figure 6: Example of correct transliterations discovered by the algorithm.

In keeping with our objective to provide as little language knowledge as possible, we introduced a simplistic approach to identifying transliteration equivalence classes, which sometimes produced erroneous groupings (e.g. an equivalence class for NE *lincoln* in Russian included both *lincoln* and *lincolnshire* on Figure 6). This approach is specific to Russian morphology, and would have to be altered for other languages. For example, for Arabic, a small set of prefixes can be used to group most NE variants. We expect that language specific knowledge used to discover accurate equivalence classes would result in performance improvements.

6 Future Work

In this work, we only consider single word Named Entities. A subject of future work is to extend the algorithm to the multi-word setting. Many of the multi-word NEs are translated as well as transliterated. For example, *Mount* in *Mount Rainier* will probably be translated, and *Rainier* - transliterated. If a dictionary exists for the two languages, it can be consulted first, and, if a match is found, transliteration model can be bypassed.

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

It seems plausible to suppose that phonetic features (if available) would help learning our transliteration model. We would like to verify if this is indeed the case.

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.

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

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

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

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

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

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

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

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

Xin Li, Paul Morie, and Dan Roth. 2004. Identification and tracing of ambiguous names: Discriminative and generative approaches. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 419–424.

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

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

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

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

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

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

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