Using Wikipedia for Cross-language Named Entity Recognition

Eraldo R. Fernandes†, Ulf Brefeld‡, Roi Blanco*, and Jordi Atserias◊

†Universidade Federal de Mato Grosso do Sul, Campo Grande, Brazil
‡Leuphana University of Lüneburg, Germany
*Yahoo! Labs, London, UK
◊University of the Basque Country, Donostia, Spain

Abstract. Named entity recognition and classification (NERC) is fundamental for natural language processing tasks such as information extraction, question answering, and topic detection. State-of-the-art NERC systems are based on supervised machine learning and hence need to be trained on (manually) annotated corpora. However, annotated corpora hardly exist for non-standard languages and labeling additional data manually is tedious and costly. In this article, we present a novel method to automatically generate (partially) annotated corpora for NERC by exploiting the link structure of Wikipedia. Firstly, Wikipedia entries in the source language are labeled with the NERC tag set. Secondly, Wikipedia language links are exploited to propagate the annotations in the target language. Finally, mentions of the labeled entities in the target language are annotated with the respective tags. The procedure results in a partially annotated corpus that is likely to contain unannotated entities. To learn from such partially annotated data, we devise two simple extensions of hidden Markov models and structural perceptrons. Empirically, we observe that using the automatically generated data leads to more accurate prediction models than off-the-shelf NERC methods. We demonstrate that the novel extensions of HMMs and perceptrons effectively exploit the partially annotated data and outperforms their baseline counterparts in all settings.

1 Introduction

The goal of named entity recognition and classification (NERC) is to detect and classify sequences of strings that represent real-world objects in natural language text. These objects are called *entities* and could for instance be mentions of people, locations, and organizations. Named entity recognition and classification is thus a fundamental component of natural language processing (NLP) pipelines and a mandatory step in many applications that deal with natural language text, including information extraction, question answering, news filtering, and topic detection and tracking [32] and has received a great deal of interest in the past years.

State-of-the-art methods for detecting entities in sentences use machine learning techniques to capture the characteristics of the involved classes of entities.

Prominent methods such as conditional random fields [23, 22] and structural support vector machines [2, 46, 48] need therefore to be adapted to annotated data before they can be deployed. Such data is for instance provided by initiatives such as CoNLL[1] that put significant effort in releasing annotated corpora for practical applications in major languages including English, German [41], Spanish, Dutch [40], Italian[2], and Chinese [49]. Although there are corpora for a few minor languages such as Catalan [27], there exist about 6,500 different languages and a large fraction thereof is not covered by NLP resources at all.

From a practitioners point of view, the performance of NERC systems highly depends on the language and the size and quality of the annotated data. If the existing resources are not sufficient for generating a model with the required predictive accuracy the data basis needs to be enlarged. However, compiling a corpus that allows to learn models with state-of-the-art performance is not only financially expensive but also time consuming as it requires manual annotations of the collected sentences. Frequently, the annotation cannot be left to laymen due to the complexity of the domain and it needs to be carried out by trained editors to deal with the pitfalls and ambiguity. In the absence of appropriate resources in the target language, the question rises whether existing corpora in another, perhaps well-studied language could be leveraged to annotate sentences in the target language. In general, cross-lingual scenarios, for instance involving parallel corpora, provide means for propagating linguistic annotations such as part-of-speech tags [51, 12], morphological information [44], and semantic roles [35]. In practice, however, creating parallel corpora is costly as, besides annotating the text, sentences need to be aligned so that translation modules can be adapted. Existing parallel corpora are therefore often small and specific in terms of the covered domain.

In this article, we study whether multilingual and freely available resources such as Wikipedia[3] can be used as surrogates to remedy the need for annotated data. Wikipedia, the largest on-line encyclopedia, has already become a widely employed resource for different NLP tasks, including Word Sense Disambiguation [30], semantic relatedness [18] or extracting semantic relationships [39]. So far, only few contributions involving Wikipedia focus on multilingual components such as cross-language question answering [16].

We present a novel approach to automatically generate (partially) annotated corpora for named entity recognition in an arbitrary language covered by Wikipedia. In the remainder, we focus on NERC and note that our approach is directly applicable to other NLP tasks such as part-of-speech tagging and word sense disambiguation. Our method comprises three stages. In the first stage, Wikipedia entries are labeled with the given NERC tag set. The second stage uses Wikipedia language links to map the entries to their peers in the target language. The third stage consists of annotating the detected entities in sentences in the target language with their corresponding tag. Note that the

[1] http://ifarm.nl/signll/conll/

[2] http://evalita.fbk.eu/

[3] http://www.wikipedia.org/

methodology leaves entities that are not linked within Wikipedia unannotated. Consequentially, the procedure results in partially labeled corpora which likely contain unannotated entities. We therefore devise two novel machine learning algorithms that are specifically tailored to process and learn from such partially annotated data based on hidden Markov models (HMMs) and structural perceptrons.

Empirically, we demonstrate that with simple extensions, machine learning algorithms are able to deal with this low-quality inexpensive data. We evaluate our approach by automatically generating mono- and cross-lingual corpora that are orders of magnitude larger than existing data sets. Empirically, we observe that using the additional data improves the performance of regular hidden Markov models and perceptrons. The novel semi-supervised algorithms significantly improve the results of their baseline counterparts by effectively exploiting the nature of the partially annotated data.

The remainder is structured as follows. Section 2 reviews related approaches to generate corpora using Wikipedia. We present the automatic generation of cross-lingual corpora using Wikipedia in Section 3 and Section 4 introduces the machine learning methods for learning from partially labeled data. We report on our empirical results in Section 5 and Section 6 concludes.

2 Wikipedia-based Corpus Generation

There are several techniques that classify Wikipedia pages using NERC labels as a preliminary step for different applications. Some of those applications include, for instance, to extend WordNet with named entities [47], to provide additional features for NERC [21], or even to classify Flickr tags [34]. However, how these techniques can be employed to generate tagged corpora is largely understudied. In the remainder of this section, we review three approaches that are related to our work.

Mika *et al.* [31] aim to improve named entity recognition and classification for English Wikipedia entries using key-value pairs of the semi-structured info boxes. Ambiguity is reduced by aggregating observed tags of tokens (the values) with respect to the fields of the info boxes (the keys). Regular expressions are used to re-label the entities. Rather than complete sentences, the final output consists of text snippets around the detected entities. Their approach ignores language links and is therefore restricted to mono-lingual scenarios.

Nothman *et al.* [33] and Richman and Schone [38] propose methods to assign NERC tags to Wikipedia entries by manually defined patterns, key phrases, and other heuristics, respectively. [38] for instance devise key phrases that serve as a simple heuristic for assigning labels to categories and observe reasonable precision by tagging categories containing the words *people from* as *person*, those including the word *company* as *organization*, and those including *country* as *location*, etc. The two approaches focus on extracting completely annotated sentences which results in two major limitations. There is the risk of erroneously annotating tokens due to overly specified rules and heuristics because sentences must be annotated completely and consequentially, large parts of the corpus are discarded because they are likely to contain false negatives (entities which are not annotated). Compared to [38], we take a different approach by only annotating entities with high confidence and leaving the remaining tokens unlabeled. By doing so, our method acts on a much larger data basis. The final models are trained on the partially annotated sentences and render the use of heuristics and manually crafted rules unnecessary.

Alternative approaches to ours are self-training or distant supervision methods. Knowledge bases like Wikipedia are used to automatically extract information (e.g,. entities) that are used to train a classifier which is then used to detect even more entities in the resource, etc. [4]. A general problem with self-training is that the initial models are trained on only a few data points and often do not generalize well. As a consequence, erroneous tags enter the training data and may dominate the whole training process.

3 Generating Annotated Corpora using Wikipedia

This section presents our approach to automatically generate (partially) annotated corpora for named entity recognition and classification. Our method exploits the link structure of Wikipedia as well as the linkage between Wikipedias in different languages. The proposed methodology consists of 3 stages. Firstly, Wikipedia entries in the source language are annotated with the respective NERC tag set (Section 3.1). Secondly, the annotated entries are projected into the target language by following cross-lingual Wikipedia links (Section 3.2). Thirdly, anchor texts in the target language linking to previously annotated entries are labeled with the corresponding tag (Section 3.3).

Figure 1 illustrates the cross-lingual setting for English (left, source language) and Spanish (right, target language). For each language, there are two entries linking to the river *Danube* (Spanish: *Danubio*). The black pointer indicates the language link from the Spanish Wikipedia page to its peer in English. To generate a corpus in Spanish using English as source language we proceed as follows. The mentions of *Danube* are tagged as *location* and propagated by links 1 and 2 to the *Danube* entry. In our simple scenario, the resulting distribution (*person*:0, *location*:2, ...) clearly indicates that this entry should be annotated as a *location.* Using the Wikipedia language link (link 3), the annotation is propagated to the Spanish entry *Danubio* which is also tagged as *location.* Finally, anchor texts in the Spanish Wikipedia of links four and five pointing to *Danubio* are accordingly annotated as locations. We obtain a partial annotation of the Spanish Wikipedia where mentions of *Danubio* are tagged as *location.*

Table 1 shows an exemplary sentence and its partial annotation. According to described procedure, *Danubio* is successfully annotated as a *location.* Words that are not linked to Wikipedia entries such as *que* as well as words that do correspond to Wikipedia entries but have not been processed yet such as *Carlomagno* remain unlabeled. Since not all entries can be resolved, the final corpus is only partially annotated.

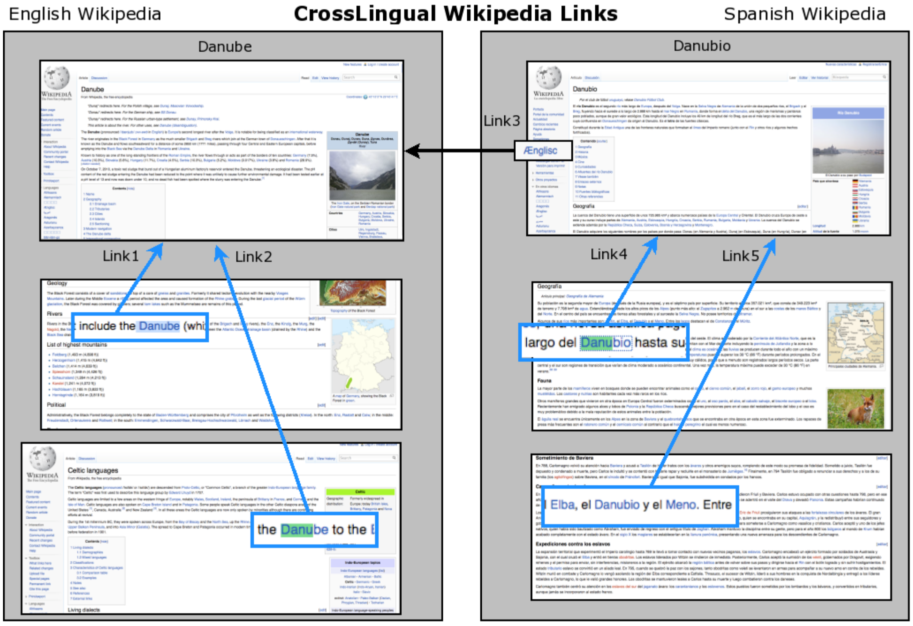

Fig. 1. Wikipedia link structure.

Table 1. A partially annotated sentence.

Carlomagno	contribuyó	a	que	el	*Danubio*	fuese	navegable
?	?	?	?	?	B-LOC	?	?

In the remainder, we focus on the English CoNLL-2003 tags, however we note that the choice of the tags is problem dependent and the incorporation of other tagsets is straight forward. The CoNLL-2003 tags are *PER* (person), *LOC* (location), *ORG* (organization), *MISC* (miscellaneous), and *O* (not an entity). In the following, we introduce our strategy in greater detail.

3.1 Annotating Wikipedia Entries

Our approach to labeling Wikipedia entries with elements of the tag set is based on existing resources. More specifically, we use the freely-available version of Wikipedia from [3], which provides a version of the English Wikipedia with several levels of semantic and syntactic annotations. These annotations have been automatically generated by state-of-the-art algorithms (see [3] for details). Moreover, this corpus preserves the Wikipedia link structure.

Note that the maximal possible number of annotations of the corpus depends on the number of links pointing to the Wikipedia entries in the source language. While some entries such as *Danube* are supported by more than 1,000 mentions

Table 2. Mismatch of annotation (center row) and linked text (bottom row) for an exemplary sentence.

···	are	of	the	Corts	of	Barcelona	from	1283
···	O	O	O	B-MISC	O	B-LOC	O	O
···	O	O	O	wiki/Corts_of_Barcelona			O	wiki/1283

others such as *Holidays with Pay (Agriculture) Convention, 1952* are almost not interlinked at all within Wikipedia. As a consequence, annotations will hardly be generated from this entry and the existing ones may be noisy due to ambiguity. However note that even if only highly interlinked entries are selected, a considerably large set of entries needs to be labeled to build a corpus with a reasonable number of annotations.

We proceed by propagating the annotations to the entries. Every tagged anchor link that is concisely tagged propagates its annotation, while mismatches between linked text and annotation are discarded. Table 2 shows an example of such a mismatch. The link *Cort_of_Barcelona* remains unused because neither *MISC* or *LOC* completely cover the linked text. Recall Figure 1 for another example. The anchor text linked to the English Wikipedia page *Danube* (links 1 and 2) is tagged as a *location* and the corresponding label *LOC* is assigned to the entry. The major advantages of this approach are twofold. First, it significantly reduces the human effort as no manual annotations are needed. Second, it does not depend on the language dependent category structure. Thus, this approach is generally applicable with any tagged subset of Wikipedia.

Table 3 shows the number of perfect matches between tags and Wikipedia links. Very frequent entries such as *Barcelona* or *Danube* are mostly tagged as locations while others like *Barcelona Olympics* do not show a clearly identifiable peak or are even associated with more than one label as for instance the entry *Barnet*. In general, there are many links to a Wikipedia page and the provided tags are not always perfect. It is thus necessary to reduce the number of ambiguous entities, that is Wikipedia entries that can be associated to more than one tag such as schools which can be either tagged as *organization* or as *location*, depending on the context.

Our approach however naturally allows for detecting ambiguous entities as all occurrences of tags are collected for an entry. Their counts simply serve as indicators to detect ambiguity. We observe a clear peak in the tag-distribution when an entity is not ambiguous; the majority of the annotations correspond to the true class of the respective entities. Thus, a simple strategy to reduce the noise of the tagging and to select a unique label for an entry is to perform a majority voting which corresponds to a *maximum-a-posteriori* prediction given the tag distribution. In practice, it is beneficial to incorporate a threshold θ to select only those Wikipedia entries that have been tagged at least θ-times and to filter entries whose ratio between the first and the second most common label is greater than α.

Table 3. Example of the different counting associated to CoNLL labels

	LOC	**PER**	**ORG**	**MISC**
Danube	1391	31	16	8
Barcelona	3,349	14	1	0
Barcelona Olympics	2	4	2	5
Barnet	33	10	74	0

Table 4 shows the label distribution for $\theta = 30$ and $\alpha = 0.4$ (Wikipedia) for 65,294 Wikipedia entries that are labeled by our method. For comparison, we include the approach by [38] (Category), which maps Wikipedia categories[4] to named entity tags. When applying this tagging technique, we use the same set of key phrases and results from [47], who labeled Wikipedia categories manually. In a few cases multiple assignments are possible; in these cases, we assign the tags to the category matching the most key phrases. Using the category strategy, we obtain NE labels for 755,770 Wikipedia entries. Note that there is no Wikipedia entry assigned to *MISC* as the original list of key phrases does not include a list for the tag *miscellaneous*. Further note that it is generally difficult to define key phrases and detection rules for inhomogeneous entity classes such as *miscellaneous* which are often inalienable in NER as they pool entities that cannot be associated with high confidence to one of the other classes. Another drawback of the category approach is that the entries are found via the Wikipedia category structure and that there is no guarantee for obtaining highly interlinked entries. Recall that the number of annotated entities in our procedure is equivalent to the number of links pointing to entries. The number of resulting annotations cannot be controlled by the category approach. For instance, the category approach leads to 13M and 800K entities for the mono-lingual English → English and the cross-language English → Spanish experiments, respectively. While our approach resulted in 19M and 1.8M entities, respectively.

Table 4. Entity distribution for our method (Wikipedia), the category approach [38] (Category), and manually labeled results from [47] (Manual).

	Wikipedia		**Category**		**Manual**	
Label	Entries	%	Entries	%	Entries	%
LOC	12,981	19.8	149,333	19,7	404	11.4
ORG	17,722	27.1	107,812	14,2	55	1.5
PER	29,723	45.5	498,625	65,9	236	6.7
MISC	4,868	7.4	-	-	-	-
O	-	-	-	-	2,822	80.2
AMB	-	-	-	-	-	-
	65,294		755,770		3,517	

[4] https://en.wikipedia.org/wiki/Help:Category

Table 5. Wikipedia cross-language links.

Links Direction	**#Links**
French→ English	730,905
English→ French	687,122
Spanish→English	480,336
English→Spanish	475,921
Catalan→English	200,090
English→Catalan	163,849
Dutch→English	167,154
English→Dutch	145,089
Icelandic→English	29,623
English→Icelandic	25,887

3.2 Cross-lingual propagation

Once the NERC tags are assigned to Wikipedia entries in the source language, we project these assignments to Wikipedia entries in the target language. This approach exploits the cross-lingual links between Wikipedias in the respective languages, provided that a Wikipedia cross-language link exists between two entries.

Note that links are not bi-directional, that is the existence of a link in one direction does not necessarily imply the existence of the opposite direction. Table 5 shows the number of language links between the English Wikipedia and some smaller Wikipedias in French (1,008,744 entries), Spanish (655,537 entries), Catalan (287,160 entries), Dutch (684,105 entries) and Icelandic (29,727 entries). Particularly for non-popular languages, the number of cross-lingual links from and to the English Wikipedia varies. Moreover, some of the links are not updated, mistyped, or use different character encodings. For instance, we are only able to map 262,489 Spanish Wikipedia entries out of the 480,336 language links to the corresponding English counterparts. The opposite direction is supported only by 160,918 entries. Nevertheless, apart from the coverage, the cross-lingual propagation can be considered as almost error-free.

3.3 Corpus Annotation

Once the tags are assigned to Wikipedia entries in the target language, the anchor text of the links pointing to tagged entries are annotated with the respective tag. We obtain a partially annotated corpus, as we have no information about annotations for text outside the entity link. The next section deals with machine learning techniques to learn from these partially annotated sentences.

4 Learning from Partially Annotated Data

Traditionally, sequence models such as hidden Markov models [36, 20] and variants thereof have been applied to label sequence learning [14] tasks. Learning

Table 6. An exemplary sentence tagged with the mentioned entities.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
$\boldsymbol{x} =$	The	Danube	is	Europe	’s	second	longest	$\cdots$
$\boldsymbol{y} =$	`O`	`LOC`	`O`	`LOC`	`O`	`O`	`O`	$\cdots$
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	

procedures for generative models adjust the parameters such that the joint likelihood of training observations and label sequences is maximized. By contrast, from an application point of view, the true benefit of a label sequence predictor corresponds to its ability to find the correct label sequence given an observation sequence. Many variants of discriminative sequence models have been explored, including maximum entropy Markov models [29], perceptron re-ranking [10, 11, 2], conditional random fields [23, 24], structural support vector machines [2, 48], and max-margin Markov models [46]. In this Section, we present extensions of hidden Markov models and perceptrons that allow for learning from partially labeled data.

A characteristic of the automatically generated data is that it might include *unannotated* entities. For instance, entity mentions may not be linked to the corresponding Wikipedia entry or do not have an associated Wikipedia entry. In cross-language scenarios, linked entries in the target sentences may not be present in the source Wikipedia and thus cannot be resolved. While labeled entities in the automatically generated data are considered ground-truth, the remaining parts of the sentence likely contain erroneous annotations and the respective tokens are thus treated as *unlabeled* rather than *not an entity.*

The following section introduces the problem setting formally. Section 4.2 and 4.3 present hidden Markov models and perceptrons for learning with partially labeled data, respectively. Section 4.6 discusses ways to parameterize the methods and Section 4.7 details their parallelization for distributed computing.

4.1 Preliminaries

The task in label sequence learning [14] is to learn a mapping from a sequential input $\boldsymbol{x} = (x_1, \ldots, x_T)$ to a sequential output $\boldsymbol{y} = (y_1, \ldots, y_T)$, where each observed token $x_t \in \Omega$ is annotated with an element of a fixed output alphabet $y_t \in \Sigma$, see Table 6 for an example. Additionally, we observe some ground-truth annotations of input $\boldsymbol{x}$ denoted by the set $\boldsymbol{z} = \{(t_j, \sigma_j)\}_{j=1}^{m}$ where $1 \leq t_j \leq T$ and $\sigma_j \in \Sigma$.

Given a sample of n pairs $(\boldsymbol{x}_1, \boldsymbol{z}_1), \ldots, (\boldsymbol{x}_n, \boldsymbol{z}_n)$ the set of labels $\boldsymbol{z}_i$ determine the learning task. If for all $\boldsymbol{z}_i = \emptyset$ holds, observations are unlabeled and the setting is called an *unsupervised* learning task. In case $|\boldsymbol{z}_i| = T_i$ for $1 \leq i \leq n$ all observed tokens are labeled and we recover the standard *supervised* scenario. If sequences are either completely annotated or completely unannotated a *semi-supervised* learning task is obtained, however, the focus of this article lies on learning with *partially annotated* data which generalizes the standard learning

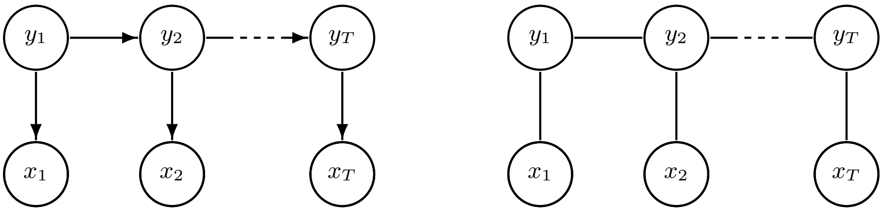

Fig. 2. Hidden Markov model (left) and Markov random field (right) for label sequence learning. The x_t denote observations and the y_i their corresponding latent variables.

tasks and does not make any assumption on the $\boldsymbol{z}_i$. In the remainder we use $\boldsymbol{x}_{[1:t]}$ as a shorthand for the sub-sequence $x_1, \ldots, x_t$ of $\boldsymbol{x}$.

4.2 Hidden Markov Models for Partially Annotated Data

We now extend hidden Markov models (HMMs) to learn from partially annotated data. The novel method combines supervised and unsupervised learning techniques for HMMs and we briefly review HMMs and the Baum-Welch algorithm in Section 4.2, respectively.

Hidden Markov Models Hidden Markov models are generative sequential models [37]. Their underlying graphical model describes how pairs $(\boldsymbol{x}, \boldsymbol{y})$ are generated and is depicted in Figure 2 (left). That is, a (first-order) hidden Markov model places an independence assumption on non-adjacent factors and computes the joint probability $P(\boldsymbol{x}, \boldsymbol{y})$ by

$$P(\boldsymbol{x}, \boldsymbol{y}) = P(y_1) \prod_{t=1}^{T} P(x_t|y_t) \prod_{t=1}^{T-1} P(y_{t+1}|y_t).$$

Priors $\pi_\sigma = P(y_1 = \sigma)$, transition probabilities $A = (a_{\sigma\tau})_{\sigma,\tau\in\Sigma}$ with $a_{\sigma\tau} = P(y_{t+1} = \tau|y_t = \sigma)$ and observation probabilities $B(\boldsymbol{x}) = (b_\sigma(x_t))_{\sigma\in\Sigma,1\leq t\leq T}$ with $b_\sigma(x_t) = P(x_t|y_t = \sigma)$ need to be adapted to the data. Usually, the parameters $\theta = (\pi, A, B)$ are estimated by maximizing the log-likelihood

$$\theta^* = \operatorname*{argmax}_\theta \sum_{i=1}^{n} \log P(\boldsymbol{x}, \boldsymbol{y}|\theta).$$

Once optimal parameters θ^* have been found, they are used as plug-in estimates to compute label distributions for unannotated sequences by means of the Forward-Backward algorithm [37]. This algorithm consists of a left-to-right pass computing $\alpha_t(\sigma)$ and a right-to-left pass that computes $\beta_t(\sigma)$. The auxiliary variables are defined as

$$\alpha_t(\sigma) = P(\boldsymbol{x}_{[1:t]}, y_t = \sigma|\theta) = \begin{cases} \pi_\sigma b_\sigma(x_1) & : \quad t = 1 \\ \sum_\tau \left[\alpha_t(\tau) a_{\tau\sigma}\right] b_\sigma(x_{t+1}) & : \quad \text{otherwise} \end{cases}$$

$$\beta_t(\sigma) = P(\boldsymbol{x}_{[t+1:T]}|y_t = \sigma, \theta) = \begin{cases} 1 & : \quad t = T \\ \sum_\tau a_{\sigma\tau} b_\tau(x_{t+1}) \beta_{t+1}(\tau) & : \quad \text{otherwise,} \end{cases}$$

and the probability for y_t taking label σ is given by

$$P(y_t = \sigma|\boldsymbol{x}, \theta) = \frac{\alpha_t(\sigma)\beta_t(\sigma)}{\sum_\tau \alpha_t(\tau)\beta_t(\tau)}.$$

Expectation Maximization In the absence of (partial) labels, that is $\bigcup \boldsymbol{z} = \emptyset$, only unlabeled input sequences $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are given. In this unsupervised case the Baum-Welch algorithm [5] is often used to learn parameters of hidden Markov models. The algorithm takes the number of possible states as input parameter and initializes the first model randomly. It then maximizes the data likelihood by an Expectation-Maximization (EM) procedure [13] consisting of two alternating steps. The *Expectation*-step computes the most likely annotations for the unlabeled sequences given the input sequences and the model. The *Maximization*-step re-estimates the model parameters given the input sequences and the previously computed annotations. The method is a variant of self-training and converges to a local optimum.

Hidden Markov Models for Partially Annotated Data We now propose an extension of hidden Markov models that learns from partially labeled data $\{(\boldsymbol{x}_i, \boldsymbol{z}_i\}_{i=1}^n$. The distribution of the labels $\boldsymbol{z}_i$ can be arbitrary and if $|\boldsymbol{z}_i| = T_i$ for all i or in case $\bigcup \boldsymbol{z}_i = \emptyset$ we recover the supervised and unsupervised hidden Markov models as special cases, respectively.

The idea is to revise the Expectation-Maximization framework as follows and grounds on the observation that annotated tokens do not need to be estimated during the *Expectation*-step. Conversely, we may use original EM updates for treating unannotated tokens as these may need to be re-estimated. This observation can be incorporated into the Forward-Backward procedure by altering the definition of the involved probabilities α and β so that the modified variants always chooses the ground-truth label for annotated tokens. The Maximization-step is identical to the original Baum-Welch algorithm but uses the modified $\tilde{\alpha}$ and $\tilde{\beta}$ variables. The modified variables are defined as

$$\tilde{\alpha}_t(\sigma) = P(\boldsymbol{x}_{[1:t]}, \boldsymbol{z}_{\leq t}, y_t = \sigma|\theta) = \begin{cases} 0 & : \quad \text{if } (t, \tau) \in \boldsymbol{z} \wedge \tau \neq \sigma \\ \alpha_t(\sigma) & : \quad \text{otherwise} \end{cases}$$

$$\tilde{\beta}_t(\sigma) = P(\boldsymbol{x}_{[t+1:T]}, \boldsymbol{z}_{>t}|y_t = \sigma, \theta) = \begin{cases} 0 & : \quad \text{if } (t, \tau) \in \boldsymbol{z}(y_t) \wedge \tau \neq \sigma \\ \beta_t(\sigma) & : \quad \text{otherwise.} \end{cases}$$

where $\boldsymbol{z}_{\leq t} = \{(t', \tau) \in \boldsymbol{z} : t' \leq t\}$ denotes the set of annotated tokens up to position t and $\boldsymbol{z}_{>t} = \boldsymbol{z} \setminus \boldsymbol{z}_{\geq t}$ are the labeled tokens at positions greater than

t. Marginalizing over the unannotated positions gives us the desired quantities; the distribution of labels at position t is for instance given by

$$P(y_t = \sigma | \boldsymbol{x}, \boldsymbol{z}, \theta) = \frac{\tilde{\alpha}_t(\sigma)\tilde{\beta}_t(\sigma)}{\sum_\tau \tilde{\alpha}_t(\tau)\tilde{\beta}_t(\tau)}.$$

The above computation schema enforces $P(y_t = \sigma | \boldsymbol{x}, \boldsymbol{z})$ for every annotated token $(t, \sigma) \in \boldsymbol{z}$ and $P(y_t = \tau | \boldsymbol{x}, \boldsymbol{z}) = 0$ for alternative labels $\tau \neq \sigma$. For unlabeled tokens x_t the original Expectation-Maximization updates are used. Note that this algorithm is a special case of [42].

4.3 Structured Perceptrons for Partially Labeled Data

The sequential learning task can alternatively be modeled in a natural way by an undirected Markov random field where we have edges between neighboring labels and between label-observation pairs, see Figure 2 (right). The conditional density $P(\boldsymbol{y}|\boldsymbol{x})$ factorizes across the cliques [19] and different feature maps can be assigned to the different types of cliques, that is ϕ_{trans} for transitions and ϕ_{obs} for emissions [2, 23]. Finally, interdependencies between $\boldsymbol{x}$ and $\boldsymbol{y}$ are captured by an aggregated joint feature map $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$,

$$\phi(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{t=2}^{T} \phi_{trans}(\boldsymbol{x}, y_{t-1}, y_t)^\top, \sum_{t=1}^{T} \phi_{obs}(\boldsymbol{x}, y_t)^\top \right)^\top .$$

We are only interested in the maximum-a-posteriori label-sequence which gives rise to log-linear models of the form

$$P(\boldsymbol{y}|\boldsymbol{x}) \propto \boldsymbol{w}^\top \phi(\boldsymbol{x}, \boldsymbol{y}).$$

The feature map exhibits a first-order Markov property and as a result, decoding can be performed by a Viterbi algorithm [17, 43] in $\mathcal{O}(T|\Sigma|^2)$,

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{w}) = \underset{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x})}{\operatorname{argmax}} \; \boldsymbol{w}^\top \phi(\boldsymbol{x}, \tilde{\boldsymbol{y}}). \quad (1)$$

In the remainder, we will focus on the 0/1- and the Hamming loss to compute the quality of predictions,

$$\ell_{0/1}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 1_{[\boldsymbol{y} \neq \hat{\boldsymbol{y}}]}; \qquad \ell_h(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{t=1}^{|\boldsymbol{y}|} 1_{[y_t \neq \hat{y}_t]} \quad (2)$$

where the indicator function $1_{[u]} = 1$ if u is true and 0 otherwise.

4.4 Loss-augmented Structured Perceptrons

The structured perceptron [10, 2] is analogous to its univariate counterpart, however, its major drawback is the minimization of the 0/1-loss which is generally too coarse for differentiating a single mislabeled token from completely erroneous annotations. To incorporate task-dependent loss functions into the learning process, we make use of the structured hinge loss of a margin-rescaled SVM [48, 28].

Given a sequence of fully labeled $(\boldsymbol{x}_1, \boldsymbol{z}_1), (\boldsymbol{x}_2, \boldsymbol{z}_2), \ldots$ where $|\boldsymbol{z}_i| = T_i$, the structured perceptron generates a sequence of models $\boldsymbol{w}_0 = \boldsymbol{0}, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots$ as follows. At time t, the loss-augmented prediction is computed by

$$\begin{aligned}\hat{\boldsymbol{y}}_t &= \underset{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x}_t)}{\operatorname{argmax}} \left[\ell(\boldsymbol{y}_t, \tilde{\boldsymbol{y}}) - \boldsymbol{w}_t^\top \phi(\boldsymbol{x}_t, \boldsymbol{y}_t) + \boldsymbol{w}_t^\top \phi(\boldsymbol{x}_t, \tilde{\boldsymbol{y}})\right] \\ &= \underset{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x}_t)}{\operatorname{argmax}} \left[\ell(\boldsymbol{y}_t, \tilde{\boldsymbol{y}}) + \boldsymbol{w}_t^\top \phi(\boldsymbol{x}_t, \tilde{\boldsymbol{y}})\right].\end{aligned}$$

Rescaling the margin with the actual loss $\ell(\boldsymbol{y}_t, \tilde{\boldsymbol{y}})$ can be intuitively motivated by recalling that the size of the margin $\gamma = \tilde{\gamma}/\|\boldsymbol{w}\|$ quantifies the confidence in rejecting an erroneously decoded output $\tilde{\boldsymbol{y}}$. Re-weighting $\tilde{\gamma}$ with the current loss $\ell(\boldsymbol{y}, \tilde{\boldsymbol{y}})$ leads to a weaker rejection confidence when $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ are similar, while large deviations from the true annotation imply a large rejection threshold. Rescaling the margin by the loss implements the intuition that the confidence of rejecting a mistaken output is proportional to its error.

An update is performed if the loss-augmented prediction $\hat{\boldsymbol{y}}_t$ does not coincide with the true output $\boldsymbol{y}_t$; the update rule is identical to that of the structured perceptron and given by

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \phi(\boldsymbol{x}_t, \boldsymbol{y}_t) - \phi(\boldsymbol{x}_t, \hat{\boldsymbol{y}}_t).$$

After an update, the model favors $\boldsymbol{y}_t$ over $\hat{\boldsymbol{y}}_t$ for the input $\boldsymbol{x}_t$, however, note that in case $\hat{\boldsymbol{y}}_t = \boldsymbol{y}_t$ the model is not changed because $\phi(\boldsymbol{x}_t, \boldsymbol{y}_t) - \phi(\boldsymbol{x}_t, \hat{\boldsymbol{y}}_t) = \boldsymbol{0}$ and thus $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t$.

Margin-rescaling can always be integrated into the decoding algorithm when the loss function decomposes over the latent variables of the output structure as it is the case for the Hamming loss in Eq. (2). After the learning process, the final model $\boldsymbol{w}^*$ is a minimizer of a convex-relaxation of the theoretical loss (the generalization error) and given by

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmin}}\ \mathbb{E}\left[\max_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x})} \ell(\boldsymbol{y}, \tilde{\boldsymbol{y}}) - \boldsymbol{w}^\top \left(\phi(\boldsymbol{x}, \boldsymbol{y}) - \phi(\boldsymbol{x}, \tilde{\boldsymbol{y}})\right)\right].$$

4.5 Transductive Perceptrons for Partially Labeled Data

We derive a straight-forward transductive extension of the loss-augmented perceptron that allows for dealing with partially annotated sequences and arbitrary (partial) labelings $\boldsymbol{z}$ [15]. The idea is to replace the missing ground-truth with a pseudo-reference labeling for incompletely annotated observation sequences. We thus propagate the fragmentary annotations to unlabeled tokens so that we obtain the desired reference labeling as a makeshift for the missing ground-truth.

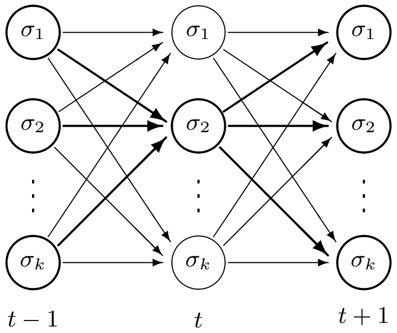

Fig. 3. The constrained Viterbi decoding (emissions are not shown). If time t is annotated with σ_2, the light edges are removed before decoding to guarantee that the optimal path passes through σ_2.

Following the transductive principle, we use a constrained Viterbi algorithm [7] to decode a pseudo ground-truth $\boldsymbol{y}_p$ for the tuple $(\boldsymbol{x}, \boldsymbol{z})$,

$$\boldsymbol{y}_p = \underset{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x})}{\operatorname{argmax}} \ \boldsymbol{w}^\top \phi(\boldsymbol{x}, \tilde{\boldsymbol{y}}) \quad \text{s.t.} \quad \forall (t, \sigma) \in \boldsymbol{z} : \quad \tilde{y}_t = \sigma.$$

The constrained Viterbi decoding guarantees that the optimal path passes through the already known labels by removing unwanted edges, see Figure 3. Assuming that a labeled token is at position $1 < t < T$, the number of removed edges is precisely $2(k^2 - (k-1)k)$, where $k = |\Sigma|$. Algorithmically, the constrained decoding splits sequences at each labeled token in two halves which are then treated independently of each other in the decoding process.

Given the pseudo labeling $\boldsymbol{y}_p$ for an observation $\boldsymbol{x}$, the update rule of the loss-augmented perceptron can be used to complement the transductive perceptron. Note that augmenting the loss function into the computation of the argmax gives $\boldsymbol{y}_p = \hat{\boldsymbol{y}}$ if and only if the prediction $\hat{\boldsymbol{y}}$ fulfills the implicit loss-rescaled margin criterion and $\phi(\boldsymbol{x}, \boldsymbol{y}_p) - \phi(\boldsymbol{x}, \hat{\boldsymbol{y}}) = \boldsymbol{0}$ holds.

Analogously to the regular perceptron algorithm, the proposed transductive generalization can easily be kernelized. Note that the weight vector at time t is given by

$$\begin{aligned} \boldsymbol{w}_t &= \boldsymbol{0} + \sum_{j=1}^{t-1} \phi(\boldsymbol{x}_j, \boldsymbol{y}_j^p) - \phi(\boldsymbol{x}_j, \tilde{\boldsymbol{y}}_j) \\ &= \sum_{(\boldsymbol{x}, \boldsymbol{y}_p, \hat{\boldsymbol{y}})} \alpha_{\boldsymbol{x}}(\boldsymbol{y}_p, \hat{\boldsymbol{y}}) \left[\phi(\boldsymbol{x}, \boldsymbol{y}_p) - \phi(\boldsymbol{x}, \hat{\boldsymbol{y}}) \right] \end{aligned} \tag{3}$$

with appropriately chosen α's that act as virtual counters, detailing how many times the prediction $\hat{\boldsymbol{y}}$ has been decoded instead of the pseudo-output $\boldsymbol{y}_p$ for an observation $\boldsymbol{x}$. Thus, the dual perceptron has virtually exponentially many parameters, however, these are initialized with $\alpha_{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{y}') = 0$ for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}'$ so that the counters only need to be instantiated once the respective triplet is actually seen. Using Eq. (3), the decision function depends only on inner products of joint feature representations which can then be replaced by appropriate kernel functions $k(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}') = \phi(\boldsymbol{x}, \boldsymbol{y})^\top \phi(\boldsymbol{x}', \boldsymbol{y}')$.

4.6 Parametrization

The presented extensions of hidden Markov models and structural perceptrons learn from labeled and unlabeled tokens. In practical applications, the unlabeled tokens usually outnumber the labeled ones and thus dominate the optimization problems and consequentially valuable label information does only have little or no impact at all on the final model. A remedy is to differently weight the influence of labeled and unlabeled data or to increase the influence of unlabeled examples during the learning process.

For the hidden Markov models, we introduce a mixing-parameter $0 \leq \lambda \leq 1$ to balance the contribution of labeled and unlabeled tokens such that the final model can be written as

$$HMM_{final}(\mathcal{D}) = (1 - \lambda) HMM_S(\mathcal{D}_{\mathcal{L}}) + \lambda HMM_U(\mathcal{D}_U),$$

where $\mathrm{HMM}_S(\mathcal{D}_L)$ and $\mathrm{HMM}_U(\mathcal{D}_U)$ correspond to supervised (HMM_S) and unsupervised (HMM_U) HMMs which are solely trained on the labeled part $\mathcal{D}_L$ and unlabeled part $\mathcal{D}_U$ of the data $\mathcal{D} = \mathcal{D}_L \bigcup \mathcal{D}_U$, respectively. For the perceptrons, we parameterize the Hamming loss to account for labeled and unlabeled tokens,

$$\ell_h(\boldsymbol{y}_p, \hat{\boldsymbol{y}}) = \sum_{t=1}^{|\boldsymbol{y}_p|} \lambda(\boldsymbol{z}, t) 1_{[y_t^p \neq \hat{y}_t]}$$

where $\lambda(\boldsymbol{z}, t) = \lambda_L$ if t is a labeled time slice, that is $(t, \cdot) \in \boldsymbol{z}$, and $\lambda(\boldsymbol{z}, t) = \lambda_U$ otherwise. Appropriate values of λ_{HMM}, λ_L and λ_U can be found using cross-validation or using holdout data.

4.7 Distributed Model Generation

The discussed hidden Markov models and perceptrons can easily be distributed on several machines. For instance, EM-like algorithms process training instances one after another and store tables with counts for each instance in the Estimation-step. The counting can be performed on several machines in parallel as the tables can easily be merged in a single process before the Maximization-step which is again a single process. After the optimization, the actual model is distributed across the grid for the next Expectation-step.

Perceptron-like algorithms can be distributed by using the results by Zinkevich et al. [53]. The idea is similar to that of EM-like algorithms. Equation (3) shows that the order of the training examples is not important as long as

Table 7. Descriptive statistics for the English → English corpora.

	CoNLL	**Wikipedia**
Tokens	203,621	1,205,137,774
Examples	14,041	57,113,370
Tokens per example	14.5	21.10
Entities	23,499	19,364,728
Entities per example	1.67	0.33
Examples with entity	79.28%	21.28%
MISC entities	14.63%	13.78%
PER entities	28.08%	29.95%
ORG entities	26.89%	32.80%
LOC entities	30.38%	23.47%

counters store the number of times they have been used for updates. Thus, the model generation can be distributed across machines and a final merging process computes the joint model which is then distributed across the grid for the next iteration.

4.8 Related Work

Learning with partially labeled data generalizes semi-supervised learning [8] which aims at reducing the need for large annotated corpora by incorporating unlabeled examples in the optimization. Semi-supervised structural prediction models have been proposed in the literature by means of Laplacian priors [24, 1], entropy-based criteria [25], transduction [52], co-training [6], self-training [26], or SDP relaxations [50]. Although these methods have been shown to improve over the performance of purely supervised structured baselines, they do not reduce the amount of required labeled examples significantly as it is sometimes the case for univariate semi-supervised learning. One of the key reasons is the variety and number of possible annotations for the same observation sequence; there are $|\Sigma|^T$ different annotations of a sequence of length T with tag set Σ and many of them are similar to each other in the sense that they differ only in a few labels. Furthermore, the above mentioned methods hardly scale for Wikipedia-sized data sets. Thus the closest method to the proposed extension of the structural perceptron is [45]. Both approaches rely on the same underlying graphical model and types of features, and use EM-like optimization strategies. We thus consider them as of the same family of approaches and note that our approach is conceptionally simpler than the one presented in [45].

5 Evaluation

In this section we report on our empirical evaluation of the automatic corpus generation. The remainder of this section is organized as follows. Section 5.1 summarizes the CoNLL data sets and Section 5.2 details our experimental setup.

We report on mono-lingual results for English in Section 5.3 and summarize the cross-lingual experiments in Section 5.4.

5.1 CoNLL Corpora

We use the English, Spanish and Dutch versions of Wikipedia to evaluate our system since manually annotated corpora are available for these languages. We use the corpora provided by CoNLL shared tasks in 2002 [40] and 2003 [41]. The CoNLL'2003 shared task [41] corpus for English includes annotations of four types of entities: person (PER), organization (ORG), location (LOC), and miscellaneous (MISC). This corpus is assembled of *Reuters News*[5] stories and divided into three parts: 203,621 training, 51,362 development, and 46,435 test tokens.

In the CoNLL'2002 shared task for Spanish and Dutch, entities are annotated using the same directives as in the English CoNLL'2003 corpus and hence comprise the same four types. The Spanish CoNLL'2002 corpus [40] consists of news wire articles from the *EFE*[6] news agency and is divided into 273,037 training, 54,837 development and 53,049 test tokens. The Dutch CoNLL'2002 corpus [40] consists of four editions of the Belgian newspaper "De Morgen" from the year 2000. The data was annotated as part of the Atranos project at the University of Antwerp. The corpus consists of 202,644 training, 37,687 development and 68,875 test tokens.

Our cross-language scenarios are based on tagged versions of Wikipedia. For English, we use the freely available resource provided by [3] as a starting point while for Spanish, we tagged the complete Spanish Wikipedia using a classifier based on the supersense tagger (SST) [9].

5.2 Experimental Setup

We use the original split of the CoNLL corpora into training, development, and test data, where the development data is used for model selection. In each experiment we compare traditional hidden Markov models and structural loss-augmented perceptrons with their extensions for learning from partially labeled data, respectively, as introduced in Section 4. The baselines are trained on the CoNLL training sets in the target language while their extensions additionally incorporate the automatically labeled data into the training processes. Perceptrons use 4 different groups of features, the word itself, its stem, part-of-speech, and shape/surface-clues. Features are encoded using a hash function with 18 bits, allowing for a maximum of 2^{18} dimensions. We report on averages over 10 repetitions where performance is computed on the respective CoNLL test split in the target language.

[5] http://www.reuters.com/

[6] http://efe.com/

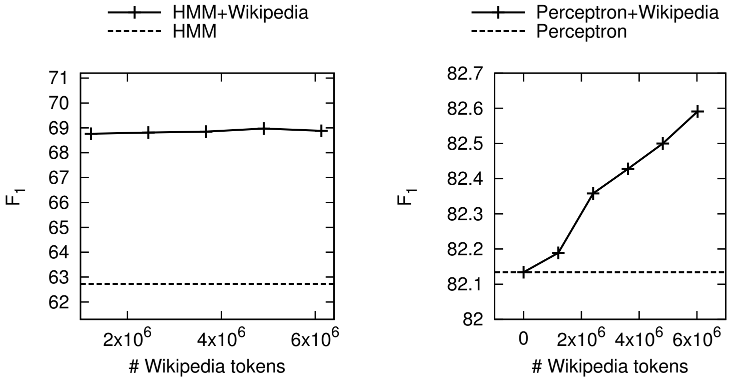

Fig. 4. Performance for English → English: HMM (left) and Perceptron (right).

5.3 Mono-language: English → English

The goal of the mono-lingual experiment is to study an ideal scenario where every entity is trivially mapped to itself instead of applying the cross-language scenario. By doing so, every detected entity is preserved and does not have to be discarded because of missing language links. Table 7 shows some descriptive statistics of the obtained corpus. As expected, entity annotations are sparser in the automatically generated corpora compared to the CoNLL training set because of false negatives as the automatically generated corpus is only partially labeled.

Using our procedure we obtain an automatically generated corpus that is about 6,000 times larger than the CoNLL training set. To assess the importance of the size of the additional sample, we randomly sample the generated corpus into smaller subsets.

Figure 4 shows the F1 performance for varying sizes of additional Wikipedia data for hidden Markov models (left) and structural perceptrons (right). Both algorithms perform significantly better than their traditional counterparts. The HMM+Wikipedia however cannot benefit from an increasing number of additional sentences due to the limited feature representation by point-distributions. By contrast, the Wikipedia enhanced perceptron uses a much richer set of features and clearly improves its performance in terms of the number of available additional data. The improvement is marginal but significant.

5.4 Cross-language Experiments

In this section, we present results on the cross-language experiments, English → Spanish, English → Dutch and finally Spanish → English. The data generation follows the protocol described in Section 3.

Table 8 compares the CoNLL and the automatically generated corpora from Wikipedia for Dutch and Spanish. As before, the generated corpora are several orders of magnitude larger than their CoNLL counterparts in all respects,

Table 8. Descriptive statistics for the English → Spanish and English → Dutch corpora

	Spanish		Dutch	
	CoNLL	**Wikipedia**	**CoNLL**	**Wikipedia**
Tokens	264,715	257,736,886	202,644	139,404,668
Examples	8,323	10,995,693	15,806	8,399,068
Tokens per example	31.81	23.44	12.82	16.60
Entities	18,798	1,837,015	13,344	8,578,923
Entities per example	2.26	0.17	0.84	1.02
Examples with entity	74.48%	12.15%	46,49 %	46.22%
MISC	11.56%	10.59%	25.01%	16.16%
PER	22.99%	35.56%	35.35%	12.23%
ORG	39.31%	23.15%	15.60%	50.48%
LOC	26.14%	30.70%	24.04%	21.13%

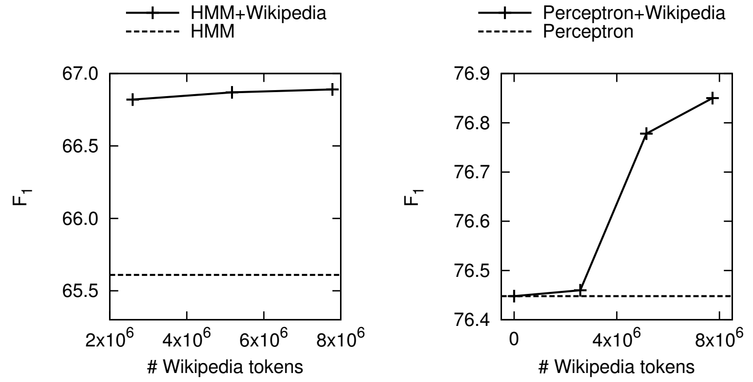

Fig. 5. Performance for English → Spanish: HMM (left) and Perceptron (right).

ranging from the number of tokens and examples to NERC annotations. Interestingly, the Spanish data has fewer entities per example and a slightly different NERC distribution than Dutch which shows a larger difference in the NERC label distribution.

Figure 5 shows our empirical findings for the cross-language scenario English → Spanish. Although the differences are not as striking as in the mono-lingual experiment, the results reflect the same trend. Again, both Wikipedia enhanced methods consistently outperform the regular HMMs and perceptrons. While the HMM+Wikipedia hardly benefits from adding more partially labeled data, the performance of the perceptron+Wikipedia jumps for $4 \times 10e6$ additional tokens; the absolute increase is again marginal but significant.

Figure 6 details results for cross-language from English to Dutch. While the HMM shows a similar behavior as for English to Spanish, the perceptron clearly suffers from including too many unlabeled examples. The last experiment studies the cross-language scenario from Spanish to English. Since English is the biggest Wikipedia and English NLP tools are usually more accurate, us-

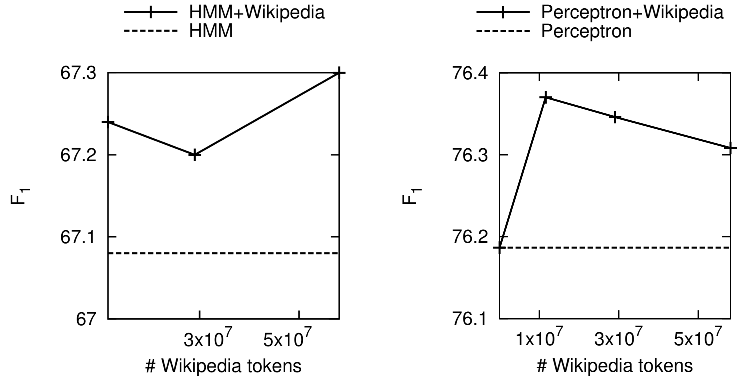

Fig. 6. Performance for English → Dutch: HMM (left) and Perceptron (right).

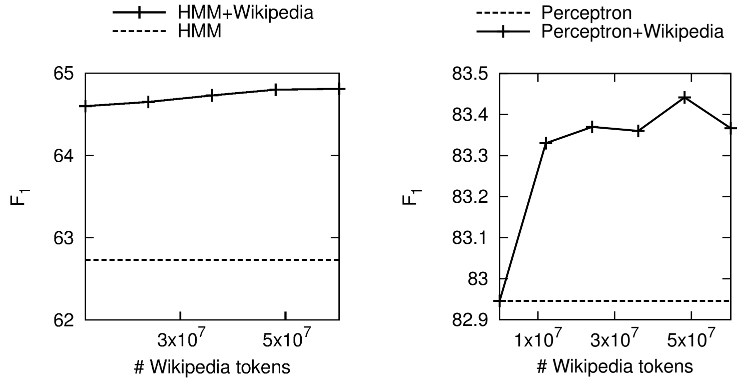

Fig. 7. Performance for Spanish → English: HMM (left) and Perceptron (right).

ing English Wikipedia as the source language seems to be a natural choice for cross-lingual NERC. Nevertheless, Figure 7 shows the results for the uncommon Spanish → English setting.

Both methods perform as expected and exhibit the already observed slight increase in performance when more partially labeled data is added. While the HMMs are clearly outperformed by the ones trained on the mono-lingual English → English data, the perceptron surprisingly increases the performance of its single-language peer. We assume that the Wikipedia language links act like a filter for ambiguous entities so that the final bi-lingual corpus contains less noise than the mono-language data. As a consequence, the corpus generated by the cross-language approach reflects the true distribution of entities in English better than the mono-lingual counterpart where every single entity is preserved.

Table 9. Descriptive statistics for the Spanish → English corpora.

	CoNLL	**Wikipedia**
Tokens	203,621	1,205,137,774
Examples	14,041	57,113,370
Tokens per example	14.5	21.10
Entities	23,499	11,917,106
Entities per example	1.67	0.67
Examples with entity	79.28%	41.65%
MISC entities	14.63%	19.42%
PER entities	28.08%	12.86%
ORG entities	26.89%	33.87%
LOC entities	30.38%	33.85%

6 Conclusions

We studied cross-language named entity recognition and classification (NERC) and presented an automatic approach to generate partially annotated corpora automatically from Wikipedia. Our method consisted of three stages. Firstly, we assigned the NERC tags to Wikipedia entries in the source language. Secondly, we exploited Wikipedia language links to translate entries into the desired target language. Thirdly, we generated a partially labeled corpus by annotating sentences from Wikipedia in the target language.

We devised simple extensions of hidden Markov models and loss-augmented perceptrons to learn from the partially annotated data. The data generation as well as the proposed extensions to the traditional learning algorithms were orthogonal to state-of-the-art approaches and could be easily included in any structural prediction model such as structural support vector machines and conditional random fields.

Our empirical results showed that using the automatically generated corpus as additional data is beneficial and leads to more accurate predictions than off-the-shelf methods. The observed improvements in performance were marginal but significant. We remark that NERC is mandatory for high-level text processing and that small improvements might have a large impact on higher-level applications as errors accumulate across the processing pipeline.

Future work will extend the presented figures with results for more partially labeled data and address the impact of the number of cross-language links of Wikipedia and the assignment of the labels to Wikipedia entries. We also intend to exploit the context of Wikipedia entities given by the link structure as an alternative denoising step. Although we focused on NERC as underlying task, our approach is generally applicable and can be straight-forwardly adapted to other NLP tasks including word sense disambiguation and part-of-speech tagging so that another interesting line of research is to extend our method to other sequential tasks.

References

1. Y. Altun, D. McAllester, and M. Belkin. Maximum margin semi–supervised learning for structured variables. In *Advances in Neural Information Processing Systems*, 2006.
2. Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proceedings of the International Conference on Machine Learning*, 2003.
3. Jordi Atserias, Hugo Zaragoza, Massimiliano Ciaramita, and Giuseppe Attardi. Semantically annotated snapshot of the english wikipedia. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.
4. Isabelle Augenstein, Diana Maynard, and Fabio Ciravegna. Relation extraction from the web using distant supervision. In *Proceedings of the International Conference on Knowledge Engineering and Knowledge Management*, 2014.
5. Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970.
6. U. Brefeld and T. Scheffer. Semi-supervised learning for structured output variables. In *Proceedings of the International Conference on Machine Learning*, 2006.
7. L. Cao and C. W. Chen. A novel product coding and recurrent alternate decoding scheme for image transmission over noisy channels. *IEEE Transactions on Communications*, 51(9):1426 – 1431, 2003.
8. O. Chapelle, B. Schölkopf, and A. Zien. *Semi–supervised Learning.* MIT Press, 2006.
9. M. Ciaramita and Y. Altun. Broad-coverage sense disambiguation and information extraction with a supersense sequence tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.
10. M. Collins. Discriminative reranking for natural language processing. In *Proceedings of the International Conference on Machine Learning*, 2000.
11. M. Collins. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.
12. S. Cucerzan and D. Yarowsky. Bootstrapping a multilingual part-of-speech tagger in one person-day. In *Proceedings of CoNLL 2002*, pages 132–138, 2002.
13. A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
14. T.G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, 2002.
15. E. R. Fernandes and U. Brefeld. Learning from partially annotated sequences. In *Proceedings of the European Conference on Machine Learning*, 2011.
16. Sergio Ferrández, Antonio Toral, Óscar Ferrández, Antonio Ferrández, and Rafael Muñoz. Exploiting wikipedia and eurowordnet to solve cross-lingual question answering. *Inf. Sci.*, 179(20):3473–3488, 2009.
17. G. D. Forney. The Viterbi algorithm. *Proceedings of IEEE*, 61(3):268–278, 1973.
18. Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In Manuela M. Veloso, editor, *IJCAI*, pages 1606–1611, 2007.

19. J. M. Hammersley and P. E. Clifford. Markov random fields on finite graphs and lattices. Unpublished manuscript, 1971.
20. B. Juang and L. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33:251–272, 1991.
21. Jun'ichi Kazama and Kentaro Torisawa. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
22. J. Lafferty, Y. Liu, and X. Zhu. Kernel conditional random fields: Representation, clique selection, and semi–supervised learning. Technical Report CMU-CS-04-115, School of Computer Science, Carnegie Mellon University, 2004.
23. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.
24. J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *Proceedings of the International Conference on Machine Learning*, 2004.
25. C. Lee, S. Wang, F. Jiao, R. Greiner, and D. Schuurmans. Learning to model spatial dependency: Semi–supervised discriminative random fields. In *Advances in Neural Information Processing Systems*, 2007.
26. W. Liao and S. Veermamachaneni. A simple semi-supervised algorithm for named entity recognition. In *Proceedings of the NAACL HLT Workshop on Semi-supervised Learning for Natural Language Processing*, 2009.
27. Lluís Màrquez, Adrià de Gispert, Xavier Carreras, and Lluís Padró. Low-cost named entity classification for catalan: Exploiting multilingual resources and unlabeled data. In *Proceedings of the ACL 2003 Workshop on Multilingual and Mixed-language Named Entity Recognition*, pages 25–32, Sapporo, Japan, July 2003. Association for Computational Linguistics.
28. D. McAllester, T. Hazan, and J. Keshet. Direct loss minimization for structured prediction. In *Advances in Neural Information Processing Systems*, 2010.
29. A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning*, 2000.
30. Rada Mihalcea. Using wikipedia for automatic word sense disambiguation. In *Proceedings of NAACL HLT 2007*, pages 196–203, 2007.
31. Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias. Learning to tag and tagging to learn: A case study on wikipedia. *IEEE Intelligent Systems*, 23:26–33, 2008.
32. David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.
33. Joel Nothman, Tara Murphy, and James R. Curran. Analysing wikipedia and gold-standard corpora for ner training. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620, Morristown, NJ, USA, 2009. Association for Computational Linguistics.
34. Simon Overell, Börkur Sigurbjörnsson, and Roelof van Zwol. Classifying tags using open content resources. In *WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 64–73, New York, NY, USA, 2009. ACM.

35. Sebastian Padó and Mirella Lapata. Cross-linguistic projection of role-semantic information. In *HLT/EMNLP*. The Association for Computational Linguistics, 2005.
36. L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.
37. Lawrence R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
38. Alexander E. Richman and Patrick Schone. Mining wiki resources for multilingual named entity recognition. In *Proceedings of ACL-08: HLT*, pages 1–9, Columbus, Ohio, June 2008. Association for Computational Linguistics.
39. Maria Ruiz-casado, Enrique Alfonseca, and Pablo Castells. Automatising the learning of lexical patterns: an application to the enrichment of wordnet by extracting semantic relationships from wikipedia. *Journal of Data and Knowledge Engineering*, 61:484–499, 2007.
40. Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: language-independent named entity recognition. In *COLING-2002: proceedings of the 6th conference on Natural language learning*, pages 1–4, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
41. Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, 2003.
42. T. Scheffer and S. Wrobel. Active hidden Markov models for information extraction. In *Proceedings of the International Symposium on Intelligent Data Analysis*, 2001.
43. R. Schwarz and Y. L. Chow. The n-best algorithm: An efficient and exact procedure for finding the n most likely hypotheses. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1990.
44. Benjamin Snyder and Regina Barzilay. Cross-lingual propagation for morphological analysis. In Dieter Fox and Carla P. Gomes, editors, *AAAI*, pages 848–854. AAAI Press, 2008.
45. J. Suzuki and H. Isozaki. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *Proceedings of ACL-08: HLT*, 2008.
46. B. Taskar, C. Guestrin, and D. Koller. Max–margin Markov networks. In *Advances in Neural Information Processing Systems*, 2004.
47. Antonio Toral, Rafael Muñoz, and Monica Monachini. Named entity wordnet. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.
48. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
49. Youzheng Wu, Jun Zhao, Bo Xu, and Hao Yu. Chinese named entity recognition based on multiple features. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 427–434, Morristown, NJ, USA, 2005. Association for Computational Linguistics.
50. L. Xu, D. Wilkinson, F. Southey, and D. Schuurmans. Discriminative unsupervised learning of structured predictors. In *Proceedings of the International Conference on Machine Learning*, 2006.
51. David Yarowsky and Grace Ngai. Inducing multilingual pos taggers and np bracketers via robust projection across aligned corpora. In *NAACL*, 2001.

52. A. Zien, U. Brefeld, and T. Scheffer. Transductive support vector machines for structured variables. In *Proceedings of the International Conference on Machine Learning*, 2007.
53. M. Zinkevich, M. Weimer, A. Smola, and L. Li. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems 23*, 2011.