Token and Type Constraints for Cross-Lingual Part-of-Speech Tagging

Oscar Täckström[◇†*] **Dipanjan Das**[‡] **Slav Petrov**[‡] **Ryan McDonald**[‡] **Joakim Nivre**[†*]

[◇] Swedish Institute of Computer Science
[†] Department of Linguistics and Philology, Uppsala University
[‡] Google Research, New York

`oscar@sics.se`
`{dipanjand|slav|ryanmcd}@google.com`
`joakim.nivre@lingfil.uu.se`

Abstract

We consider the construction of part-of-speech taggers for resource-poor languages. Recently, manually constructed tag dictionaries from Wiktionary and dictionaries projected via bitext have been used as *type constraints* to overcome the scarcity of annotated data in this setting. In this paper, we show that additional *token constraints* can be projected from a resource-rich source language to a resource-poor target language via word-aligned bitext. We present several models to this end; in particular a partially observed conditional random field model, where coupled token and type constraints provide a partial signal for training. Averaged across eight previously studied Indo-European languages, our model achieves a 25% relative error reduction over the prior state of the art. We further present successful results on seven additional languages from different families, empirically demonstrating the applicability of coupled token and type constraints across a diverse set of languages.

1 Introduction

Supervised part-of-speech (POS) taggers are available for more than twenty languages and achieve accuracies of around 95% on in-domain data (Petrov et al., 2012). Thanks to their efficiency and robustness, supervised taggers are routinely employed in many natural language processing applications, such as syntactic and semantic parsing, named-entity recognition and machine translation. Unfortunately, the resources required to train supervised taggers are expensive to create and unlikely to exist for the majority of written languages. The necessity of building NLP tools for these resource-poor languages has been part of the motivation for research on unsupervised learning of POS taggers (Christodoulopoulos et al., 2010).

In this paper, we instead take a *weakly* supervised approach towards this problem. Recently, learning POS taggers with *type*-level tag dictionary constraints has gained popularity. Tag dictionaries, noisily projected via word-aligned bitext, have bridged the gap between purely unsupervised and fully supervised taggers, resulting in an average accuracy of over 83% on a benchmark of eight Indo-European languages (Das and Petrov, 2011). Li et al. (2012) further improved upon this result by employing Wiktionary[1] as a tag dictionary source, resulting in the hitherto best published result of almost 85% on the same setup.

Although the aforementioned weakly supervised approaches have resulted in significant improvements over fully unsupervised approaches, they have not exploited the benefits of *token*-level cross-lingual projection methods, which are possible with word-aligned bitext between a *target* language of interest and a resource-rich *source* language, such as English. This is the setting we consider in this paper (§2). While prior work has successfully considered both token- and type-level projection across word-aligned bitext for estimating the model parameters of generative tagging models (Yarowsky and Ngai, 2001; Xi and Hwa, 2005, *inter alia*), a key observation underlying the present work is that token- and type-level information offer different and complementary signals. On the one hand, high confidence token-level projections offer precise constraints on a tag in a particular context. On the other hand, manually cre-

[*] Work primarily carried out while at Google Research.

[1] `http://www.wiktionary.org/`.

Transactions of the Association for Computational Linguistics, 1 (2013) 1–12. Action Editor: Sharon Goldwater.
Submitted 11/2012; Revised 1/2013; Published 3/2013. ©2013 Association for Computational Linguistics.

ated type-level dictionaries can have broad coverage and do not suffer from word-alignment errors; they can therefore be used to filter systematic as well as random noise in token-level projections.

In order to reap these potential benefits, we propose a partially observed conditional random field (CRF) model (Lafferty et al., 2001) that couples token and type constraints in order to guide learning (§3). In essence, the model is given the freedom to push probability mass towards hypotheses consistent with both types of information. This approach is flexible: we can use either noisy projected or manually constructed dictionaries to generate type constraints; furthermore, we can incorporate arbitrary features over the input. In addition to standard (contextual) lexical features and transition features, we observe that adding features from a monolingual word clustering (Uszkoreit and Brants, 2008) can significantly improve accuracy. While most of these features can also be used in a generative feature-based hidden Markov model (HMM) (Berg-Kirkpatrick et al., 2010), we achieve the best accuracy with a globally normalized discriminative CRF model.

To evaluate our approach, we present extensive results on standard publicly available datasets for 15 languages: the eight Indo-European languages previously studied in this context by Das and Petrov (2011) and Li et al. (2012), and seven additional languages from different families, for which no comparable study exists. In §4 we compare various features, constraints and model types. Our best model uses type constraints derived from Wiktionary, together with token constraints derived from high-confidence word alignments. When averaged across the eight languages studied by Das and Petrov (2011) and Li et al. (2012), we achieve an accuracy of 88.8%. This is a 25% relative error reduction over the previous state of the art. Averaged across all 15 languages, our model obtains an accuracy of 84.5% compared to 78.5% obtained by a strong generative baseline. Finally, we provide an in depth analysis of the relative contributions of the two types of constraints in §5.

2 Coupling Token and Type Constraints

Type-level information has been amply used in weakly supervised POS induction, either via pure manually crafted tag dictionaries (Smith and Eisner, 2005; Ravi and Knight, 2009; Garrette and Baldridge, 2012), noisily projected tag dictionaries (Das and Petrov, 2011) or through crowdsourced lexica, such as Wiktionary (Li et al., 2012). At the other end of the spectrum, there have been efforts that project token-level information across word-aligned bitext (Yarowsky and Ngai, 2001; Xi and Hwa, 2005). However, systems that combine both sources of information in a single model have yet to be fully explored. The following three subsections outline our overall approach for coupling these two types of information to build robust POS taggers that do not require any direct supervision in the target language.

2.1 Token Constraints

For the majority of resource-poor languages, there is at least some bitext with a resource-rich source language; for simplicity, we choose English as our source language in all experiments. It is then natural to consider using a supervised part-of-speech tagger to predict part-of-speech tags for the English side of the bitext. These predicted tags can subsequently be projected to the target side via automatic word alignments. This approach was pioneered by Yarowsky and Ngai (2001), who used the resulting partial target annotation to estimate the parameters of an HMM. However, due to the automatic nature of the word alignments and the POS tags, there will be significant noise in the projected tags. To conquer this noise, they used very aggressive smoothing techniques when training the HMM. Fossum and Abney (2005) used similar token-level projections, but instead combined projections from multiple source languages to filter out random projection noise as well as the systematic noise arising from different source language annotations and syntactic divergences.

2.2 Type Constraints

It is well known that given a tag dictionary, even if it is incomplete, it is possible to learn accurate POS taggers (Smith and Eisner, 2005; Goldberg et al., 2008; Ravi and Knight, 2009; Naseem et al., 2009). While widely differing in the specific model structure and learning objective, all of these approaches achieve excellent results. Unfortunately, they rely on tag dictionaries extracted directly from the underlying treebank data. Such dictionaries provide in depth coverage of the test domain and also list all

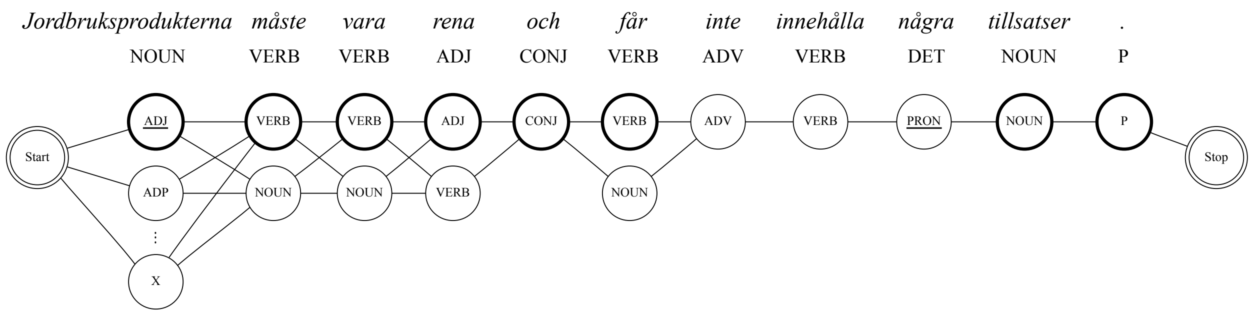

Figure 1: Lattice representation of the inference search space $\mathcal{Y}(x)$ for an authentic sentence in Swedish (*"The farming products must be pure and must not contain any additives"*), after pruning with Wiktionary type constraints. The correct parts of speech are listed underneath each word. Bold nodes show projected token constraints $\tilde{y}$. Underlined text indicates incorrect tags. The coupled constraints lattice $\widehat{\mathcal{Y}}(x, \tilde{y})$ consists of the bold nodes together with nodes for words that are lacking token constraints; in this case, the coupled constraints lattice thus defines exactly one valid path.

inflected forms – both of which are difficult to obtain and unrealistic to expect for resource-poor languages.

In contrast, Das and Petrov (2011) automatically create type-level tag dictionaries by aggregating over projected token-level information extracted from bitext. To handle the noise in these automatic dictionaries, they use label propagation on a similarity graph to smooth (and also expand) the label distributions. While their approach produces good results and is applicable to resource-poor languages, it requires a complex multi-stage training procedure including the construction of a large distributional similarity graph.

Recently, Li et al. (2012) presented a simple and viable alternative: crowdsourced dictionaries from Wiktionary. While noisy and sparse in nature, Wiktionary dictionaries are available for 170 languages.[2] Furthermore, their quality and coverage is growing continuously (Li et al., 2012). By incorporating type constraints from Wiktionary into the feature-based HMM of Berg-Kirkpatrick et al. (2010), Li et al. were able to obtain the best published results in this setting, surpassing the results of Das and Petrov (2011) on eight Indo-European languages.

2.3 Coupled Constraints

Rather than relying exclusively on either token or type constraints, we propose to complement the one with the other during training. For each sentence in our training set, a partially constrained lattice of tag sequences is constructed as follows:

1. For each token whose type is *not* in the tag dictionary, we allow the entire tag set.
2. For each token whose type *is* in the tag dictionary, we prune all tags not licensed by the dictionary and mark the token as dictionary-pruned.
3. For each token that has a tag projected via a high-confidence bidirectional word alignment: if the projected tag is still present in the lattice, then we prune every tag but the projected tag for that token; if the projected tag is not present in the lattice, which can only happen for dictionary-pruned tokens, then we ignore the projected tag.

Figure 1 provides a running example. The lattice shows tags permitted after constraining the words to tags licensed by the dictionary (up until Step 2 from above). There is only a single token "Jordbruksprodukterna" (*"the farming products"*) not in the dictionary; in this case the lattice permits the full set of tags. With token-level projections (Step 3; nodes with bold border in Figure 1), the lattice can be further pruned. In most cases, the projected tag is both correct and is in the dictionary-pruned lattice. We thus successfully disambiguate such tokens and shrink the search space substantially.

There are two cases we highlight in order to show where our model can break. First, for the token "Jordbruksprodukterna", the erroneously projected tag ADJ will eliminate all other tags from the lattice, including the correct tag NOUN. Second, the token "några" (*"any"*) has a single dictionary entry PRON and is missing the correct tag DET. In the case where

[2] http://meta.wikimedia.org/wiki/Wiktionary — October 2012.

DET is the projected tag, we will not add it to the lattice and simply ignore it. This is because we hypothesize that the tag dictionary can be trusted more than the tags projected via noisy word alignments. As we will see in §4, taking the union of tags performs worse, which supports this hypothesis.

For *generative* models, such as HMMs (§3.1), we need to define only one lattice. For our best generative model this is the coupled token- and type-constrained lattice.[3] At prediction time, in both the discriminative and the generative cases, we find the most likely label sequence using Viterbi decoding.

For *discriminative* models, such as CRFs (§3.2), we need to define two lattices: one that the model moves probability mass towards and another one defining the overall search space (or partition function). In traditional supervised learning without a dictionary, the former is a trivial lattice containing the gold standard tag sequence and the latter is the set of all possible tag sequences spanning the tokens. With our best model, we will move mass towards the coupled token- and type-constrained lattice, such that the model can freely distribute mass across all paths consistent with these constraints. The lattice defining the partition function will be the full set of possible tag sequences when no dictionary is used; when a dictionary is used it will consist of all dictionary-pruned tag sequences (sans Step 3 above; the full set of possibilities shown in Figure 1 for our running example).

Figures 2 and 3 provide statistics regarding the supervision coverage and remaining ambiguity. Figure 2 shows that more than two thirds of all tokens in our training data are in Wiktionary. However, there is considerable variation between languages: Spanish has the highest coverage with over 90%, while Turkish, an agglutinative language with a vast number of word forms, has less than 50% coverage. Figure 3 shows that there is substantial uncertainty left after pruning with Wiktionary, since tokens are rarely fully disambiguated: 1.3 tags per token are allowed on average for types in Wiktionary.

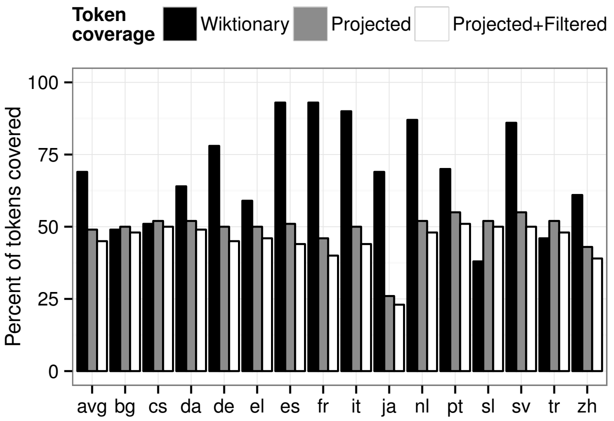

Figure 2: Wiktionary and projection dictionary coverage. Shown is the percentage of tokens in the target side of the bitext that are covered by Wiktionary, that have a projected tag, and that have a projected tag after intersecting the two.

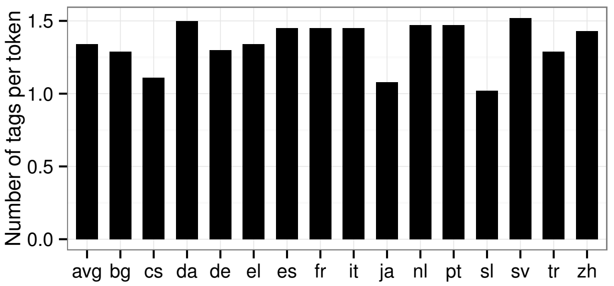

Figure 3: Average number of licensed tags per token on the target side of the bitext, for types in Wiktionary.

Figure 2 further shows that high-confidence alignments are available for about half of the tokens for most languages (Japanese is a notable exception with less than 30% of the tokens covered). Intersecting the Wiktionary tags and the projected tags (Step 2 and 3 above) filters out some of the potentially erroneous tags, but preserves the majority of the projected tags; the remaining, presumably more accurate projected tags cover almost half of all tokens, greatly reducing the search space that the learner needs to explore.

3 Models with Coupled Constraints

We now formally present how we couple token and type constraints and how we use these coupled constraints to train probabilistic tagging models. Let $x = (x_1 x_2 \dots x_{|x|}) \in \mathcal{X}$ denote a sentence, where each token $x_i \in \mathcal{V}$ is an instance of a word type from the vocabulary $\mathcal{V}$ and let $y = (y_1 y_2 \dots y_{|x|}) \in \mathcal{Y}$ denote a tag sequence, where $y_i \in \mathcal{T}$ is the tag assigned to token x_i and $\mathcal{T}$ denotes the set of all possible part-of-speech tags. We denote the lattice of all admissible tag sequences for the sentence x by $\mathcal{Y}(x)$. This is the

[3] Other training methods exist as well, for example, contrastive estimation (Smith and Eisner, 2005).

inference search space in which the tagger operates. As we shall see, it is crucial to constrain the size of this lattice in order to simplify learning when only incomplete supervision is available.

A tag dictionary maps a word type $x_j \in \mathcal{V}$ to a set of admissible tags $\mathcal{T}(x_j) \subseteq \mathcal{T}$. For word types not in the dictionary we allow the full set of tags $\mathcal{T}$ (while possible, in this paper we do not attempt to distinguish closed-class versus open-class words). When provided with a tag dictionary, the lattice of admissible tag sequences for a sentence x is $\mathcal{Y}(x) = \mathcal{T}(x_1) \times \mathcal{T}(x_2) \times \ldots \times \mathcal{T}(x_{|x|})$. When no tag dictionary is available, we simply have the full lattice $\mathcal{Y}(x) = \mathcal{T}^{|x|}$.

Let $\tilde{y} = (\tilde{y}_1 \tilde{y}_2 \ldots \tilde{y}_{|x|})$ be the projected tags for the sentence x. Note that $\{\tilde{y}_i\} = \emptyset$ for tokens without a projected tag. Next, we define a piecewise operator $\frown$ that couples $\tilde{y}$ and $\mathcal{Y}(x)$ with respect to every sentence index, which results in a token- and type-constrained lattice. The operator behaves as follows, coherent with the high level description in §2.3:

$$\widehat{\mathcal{T}}(x_i, \tilde{y}_i) = \tilde{y}_i \frown \mathcal{T}(x_i) = \begin{cases} \{\tilde{y}_i\} & \text{if } \tilde{y}_i \in \mathcal{T}(x_i) \\ \mathcal{T}(x_i) & \text{otherwise}. \end{cases}$$

We denote the token- and type-constrained lattice as $\widehat{\mathcal{Y}}(x, \tilde{y}) = \widehat{\mathcal{T}}(x_1, \tilde{y}_1) \times \widehat{\mathcal{T}}(x_2, \tilde{y}_2) \times \ldots \times \widehat{\mathcal{T}}(x_{|x|}, \tilde{y}_{|x|})$. Note that when token-level projections are not used, the dictionary-pruned lattice and the lattice with coupled constraints are identical, that is $\widehat{\mathcal{Y}}(x, \tilde{y}) = \mathcal{Y}(x)$.

3.1 HMMs with Coupled Constraints

A first-order hidden Markov model (HMM) specifies the joint distribution of a sentence $x \in \mathcal{X}$ and a tag-sequence $y \in \mathcal{Y}(x)$ as:

$$p_\beta(x, y) = \prod_{i=1}^{|x|} \underbrace{p_\beta(x_i \mid y_i)}_{\text{emission}} \underbrace{p_\beta(y_i \mid y_{i-1})}_{\text{transition}} .$$

We follow the recent trend of using a log-linear parametrization of the emission and the transition distributions, instead of a multinomial parametrization (Chen, 2003). This allows model parameters β to be shared across categorical events, which has been shown to give superior performance (Berg-Kirkpatrick et al., 2010). The categorical emission and transition events are represented by feature vectors $\phi(x_i, y_i)$ and $\phi(y_i, y_{i-1})$. Each element of the parameter vector β corresponds to a particular feature; the component log-linear distributions are:

$$p_\beta(x_i \mid y_i) = \frac{\exp\left(\beta^\top \phi(x_i, y_i)\right)}{\sum_{x'_i \in \mathcal{V}} \exp\left(\beta^\top \phi(x'_i, y_i)\right)} ,$$

and

$$p_\beta(y_i \mid y_{i-1}) = \frac{\exp\left(\beta^\top \phi(y_i, y_{i-1})\right)}{\sum_{y'_i \in \mathcal{T}} \exp\left(\beta^\top \phi(y'_i, y_{i-1})\right)} .$$

In maximum-likelihood estimation of the parameters, we seek to maximize the likelihood of the observed parts of the data. For this we need the joint marginal distribution $p_\beta(x, \widehat{\mathcal{Y}}(x, \tilde{y}))$ of a sentence x, and its coupled constraints lattice $\widehat{\mathcal{Y}}(x, \tilde{y})$, which is obtained by marginalizing over all consistent outputs:

$$p_\beta(x, \widehat{\mathcal{Y}}(x, \tilde{y})) = \sum_{y \in \widehat{\mathcal{Y}}(x, \tilde{y})} p_\beta(x, y) .$$

If there are no projections and no tag dictionary, then $\widehat{\mathcal{Y}}(x, \tilde{y}) = \mathcal{T}^{|x|}$, and thus $p_\beta(x, \widehat{\mathcal{Y}}(x, \tilde{y})) = p_\beta(x)$, which reduces to fully unsupervised learning. The ℓ_2-regularized *marginal joint* log-likelihood of the constrained training data $\mathcal{D} = \{(x^{(i)}, \tilde{y}^{(i)})\}_{i=1}^n$ is:

$$\mathcal{L}(\beta; \mathcal{D}) = \sum_{i=1}^{n} \log p_\beta(x^{(i)}, \widehat{\mathcal{Y}}(x^{(i)}, \tilde{y}^{(i)})) - \gamma \, \|\beta\|_2^2 . \tag{1}$$

We follow Berg-Kirkpatrick et al. (2010) and take a direct gradient approach for optimizing Eq. 1 with L-BFGS (Liu and Nocedal, 1989). We set $\gamma = 1$ and run 100 iterations of L-BFGS. One could also employ the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to optimize this objective, although the relative merits of EM versus direct gradient training for these models is still a topic of debate (Berg-Kirkpatrick et al., 2010; Li et al., 2012).[4] Note that since the marginal likelihood is non-concave, we are only guaranteed to find a local maximum of Eq. 1.

After estimating the model parameters β, the tag-sequence $y^* \in \mathcal{Y}(x)$ for a sentence $x \in \mathcal{X}$ is predicted by choosing the one with maximal *joint* probability:

$$y^* \leftarrow \arg\max_{y \in \mathcal{Y}(x)} p_\beta(x, y) .$$

[4] We trained the HMM with EM as well, but achieved better results with direct gradient training and hence omit those results.

3.2 CRFs with Coupled Constraints

Whereas an HMM models the joint probability of the input $x \in \mathcal{X}$ and output $y \in \mathcal{Y}(x)$, using locally normalized component distributions, a conditional random field (CRF) instead models the probability of the output conditioned on the input as a globally normalized log-linear distribution (Lafferty et al., 2001):

$$p_\theta(y \mid x) = \frac{\exp\left(\theta^\top \Phi(x, y)\right)}{\sum_{y' \in \mathcal{Y}(x)} \exp\left(\theta^\top \Phi(x, y')\right)} \,,$$

where θ is a parameter vector. As for the HMM, $\mathcal{Y}(x)$ is not necessarily the full space of possible tag-sequences; specifically, for us, it is the dictionary-pruned lattice *without* the token constraints.

With a first-order Markov assumption, the feature function factors as:

$$\Phi(x, y) = \sum_{i=1}^{|x|} \phi(x, y_i, y_{i-1}) \,.$$

This model is more powerful than the HMM in that it can use richer feature definitions, such as joint input/transition features and features over a wider input context. We model a marginal conditional probability, given by the total probability of all tag sequences consistent with the lattice $\widehat{\mathcal{Y}}(x, \tilde{y})$:

$$p_\theta(\widehat{\mathcal{Y}}(x, \tilde{y}) \mid x) \;=\; \sum_{y \in \widehat{\mathcal{Y}}(x, \tilde{y})} p_\theta(y \mid x) \,.$$

The parameters of this constrained CRF are estimated by maximizing the ℓ_2-regularized *marginal conditional* log-likelihood of the constrained data (Riezler et al., 2002):

$$\mathcal{L}(\theta; \mathcal{D}) = \sum_{i=1}^{n} \log p_\theta(\widehat{\mathcal{Y}}(x^{(i)}, \tilde{y}^{(i)}) \mid x^{(i)}) - \gamma \|\theta\|_2^2 \,. \tag{2}$$

As with Eq. 1, we maximize Eq. 2 with 100 iterations of L-BFGS and set $\gamma = 1$. In contrast to the HMM, after estimating the model parameters θ, the tag-sequence $y^* \in \mathcal{Y}(x)$ for a sentence $x \in \mathcal{X}$ is chosen as the sequence with the maximal *conditional* probability:

$$y^* \leftarrow \arg\max_{y \in \mathcal{Y}(x)} p_\theta(y \mid x) \,.$$

4 Empirical Study

We now present a detailed empirical study of the models proposed in the previous sections. In addition to comparing with the state of the art in Das and Petrov (2011) and Li et al. (2012), we present models with several combinations of token and type constraints, additional features incorporating word clusters. Both generative and discriminative models are explored.

4.1 Experimental Setup

Before delving into the experimental details, we present our setup and datasets.

Languages. We evaluate on eight target languages used in previous work (Das and Petrov, 2011; Li et al., 2012) and on seven additional languages (see Table 1). While the former eight languages all belong to the Indo-European family, we broaden the coverage to language families more distant from the source language (for example, Chinese, Japanese and Turkish). We use the treebanks from the CoNLL shared tasks on dependency parsing (Buchholz and Marsi, 2006; Nivre et al., 2007) for evaluation.[5] The two-letter abbreviations from the ISO 639-1 standard are used when referring to these languages in tables and figures.

Tagset. In all cases, we map the language-specific POS tags to universal POS tags using the mapping of Petrov et al. (2012).[6] Since we use indirect supervision via projected tags or Wiktionary, the model states induced by all models correspond directly to POS tags, enabling us to compute tagging accuracy without a greedy 1-to-1 or many-to-1 mapping.

Bitext. For all experiments, we use English as the source language. Depending on availability, there are between 1M and 5M parallel sentences for each language. The majority of the parallel data is gathered automatically from the web using the method of Uszkoreit et al. (2010). We further include data from Europarl (Koehn, 2005) and from the UN parallel corpus (UN, 2006), for languages covered by these corpora. The English side of the bitext is POS tagged with a standard supervised CRF tagger, trained on the Penn Treebank (Marcus et al., 1993), with tags mapped to universal tags. The parallel sen-

[5] For French we use the treebank of Abeillé et al. (2003).

[6] We use version 1.03 of the mappings available at `http://code.google.com/p/universal-pos-tags/`.

tences are word aligned with the aligner of DeNero and Macherey (2011). Intersected high-confidence alignments (confidence > 0.95) are extracted and aggregated into projected type-level dictionaries. For purely practical reasons, the training data with token-level projections is created by randomly sampling target-side sentences with a total of 500K tokens.

Wiktionary. We use a snapshot of the Wiktionary word definitions, and follow the heuristics of Li et al. (2012) for creating the Wiktionary dictionary by mapping the Wiktionary tags to universal POS tags.[7]

Features. For all models, we use only an identity feature for tag-pair transitions. We use five features that couple the current tag and the observed word (analogous to the emission in an HMM): word identity, suffixes of up to length 3, and three indicator features that fire when the word starts with a capital letter, contains a hyphen or contains a digit. These are the same features as those used by Das and Petrov (2011). Finally, for some models we add a word cluster feature that couples the current tag and the word cluster identity of the word. These (monolingual) word clusters are induced with the exchange algorithm (Uszkoreit and Brants, 2008). We set the number of clusters to 256 across all languages, as this has previously been shown to produce robust results for similar tasks (Turian et al., 2010; Täckström et al., 2012). The clusters for each language are learned on a large monolingual newswire corpus.

4.2 Models with Type Constraints

To examine the sole effect of type constraints, we experiment with the HMM, drawing constraints from three different dictionaries. Table 1 compares the performance of our models with the best results of Das and Petrov (2011, D&P) and Li et al. (2012, LG&T). As in previous work, training is done exclusively on the training portion of each treebank, stripped of any manual linguistic annotation.

We first use all of our parallel data to generate projected tag dictionaries: the English POS tags are projected across word alignments and aggregated to tag distributions for each word type. As in Das and Petrov (2011), the distributions are then filtered with a threshold of 0.2 to remove noisy tags and to create an unweighted tag dictionary. We call this model $\mathcal{Y}^{\text{HMM}}_{\text{proj.}}$; its average accuracy of 84.9% on the eight languages is higher than the 83.4% of D&P and on par with LG&T (84.8%).[8] Our next model ($\mathcal{Y}^{\text{HMM}}_{\text{wik.}}$) simply draws type constraints from Wiktionary. It slightly underperforms LG&T (83.0%), presumably because they used a second-order HMM. As a simple extension to these two models, we take the union of the projected dictionary and Wiktionary to constrain an HMM, which we name $\mathcal{Y}^{\text{HMM}}_{\text{union}}$. This model performs a little worse on the eight Indo-European languages (84.5), but gives an improvement over the projected dictionary when evaluated across all 15 languages (80.0% vs. 78.5%).

	Prior work		HMM with type constraints			
Lang.	D&P	LG&T	$\mathcal{Y}^{\text{HMM}}_{\text{proj.}}$	$\mathcal{Y}^{\text{HMM}}_{\text{wik.}}$	$\mathcal{Y}^{\text{HMM}}_{\text{union}}$	$\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C
bg	–	–	84.2	68.1	87.2	**87.9**
cs	–	–	75.4	70.2	75.4	**79.2**
da	83.2	83.3	87.7	82.0	78.4	**89.5**
de	82.8	85.8	86.6	85.1	80.0	**88.3**
el	82.5	79.2	83.3	83.8	**86.0**	83.2
es	84.2	86.4	83.9	83.7	**88.3**	87.3
fr	–	–	**88.4**	75.7	75.6	86.6
it	86.8	86.5	89.0	85.4	89.9	**90.6**
ja	–	–	45.2	**76.9**	74.4	73.7
nl	79.5	**86.3**	81.7	79.1	83.8	82.7
pt	87.9	84.5	86.7	79.0	83.8	**90.4**
sl	–	–	78.7	64.8	82.8	**83.4**
sv	80.5	86.1	80.6	85.9	85.9	**86.7**
tr	–	–	**66.2**	44.1	65.1	65.7
zh	–	–	59.2	**73.9**	63.2	73.0
avg (8)	83.4	84.8	84.9	83.0	84.5	**87.3**
avg	–	–	78.5	75.9	80.0	**83.2**

Table 1: Tagging accuracies for type-constrained HMM models. D&P is the "With LP" model in Table 2 of Das and Petrov (2011), while LG&T is the "SHMM-ME" model in Table 2 of Li et al. (2012). $\mathcal{Y}^{\text{HMM}}_{\text{proj.}}$, $\mathcal{Y}^{\text{HMM}}_{\text{wik.}}$ and $\mathcal{Y}^{\text{HMM}}_{\text{union}}$ are HMMs trained solely with type constraints derived from the projected dictionary, Wiktionary and the union of these dictionaries, respectively. $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C is equivalent to $\mathcal{Y}^{\text{HMM}}_{\text{union}}$ with additional cluster features. All models are trained on the treebank of each language, stripped of gold labels. Results are averaged over the 8 languages from Das and Petrov (2011), denoted *avg (8)*, as well as over the full set of 15 languages, denoted *avg*.

[7] The definitions were downloaded on August 31, 2012 from `http://toolserver.org/~enwikt/definitions/`. This snapshot is more recent than that used by Li et al.

[8] Our model corresponds to the weaker, "No LP" projection of Das and Petrov (2011). We found that label propagation was only beneficial when small amounts of bitext were available.

		Token constraints		HMM with coupled constraints			CRF with coupled constraints		
Lang.	$\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L	$\tilde{y}^{\text{HMM}}$+C+L	$\tilde{y}^{\text{CRF}}$+C+L	$\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{proj.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{wik.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{union}}$+C+L	$\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{proj.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{wik.}}$+C+L	$\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{union}}$+C+L
bg	87.7	77.9	84.1	84.5	83.9	86.7	86.0	**87.8**	85.4
cs	78.3	65.4	74.9	74.8	**81.1**	76.9	74.7	80.3**	75.0
da	87.3	80.9	85.1	87.2	85.6	88.1	85.5	**88.2***	86.0
de	87.7	81.4	83.3	85.0	89.3	86.7	84.4	**90.5****	85.5
el	85.9	81.1	77.8	80.1	87.0	83.9	79.6	**89.5****	79.7
es	**89.1****	84.1	85.5	83.7	85.9	88.0	85.7	87.1	86.0
fr	**88.4****	83.5	84.7	85.9	86.4	87.4	84.9	87.2	85.6
it	89.6	85.2	88.5	88.7	87.6	**89.8**	88.3	89.3	89.4
ja	72.8	47.6	54.2	43.2	76.1	70.5	44.9	**81.0****	68.0
nl	83.1	78.4	82.4	82.3	84.2	83.2	83.1	**85.9****	83.2
pt	89.1	84.7	87.0	86.6	88.7	88.0	87.9	**91.0****	88.3
sl	**82.4**	69.8	78.2	78.5	81.8	80.1	79.7	82.3	80.0
sv	86.1	80.1	84.2	82.3	87.9	86.9	84.4	**88.9****	85.5
tr	62.4	58.1	64.5	64.6	61.8	64.8	65.0	64.1**	**65.2**
zh	72.6	52.7	39.5	56.0	74.1	73.3	59.7	**74.4****	73.4
avg (8)	87.2	82.0	84.2	84.5	87.0	86.8	84.9	**88.8**	85.4
avg	82.8	74.1	76.9	77.6	82.8	82.3	78.2	**84.5**	81.1

Table 2: Tagging accuracies for models with token constraints and coupled token and type constraints. All models use cluster features (…+C) and are trained on large training sets each containing 500k tokens with (partial) token-level projections (…+L). The best type-constrained model, trained on the larger datasets, $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L, is included for comparison. The remaining columns correspond to HMM and CRF models trained *only* with token constraints ($\tilde{y}\ldots$) and with coupled token and type constraints ($\widehat{\mathcal{Y}}\ldots$). The latter are trained using the projected dictionary ($\cdot_{\text{proj.}}$), Wiktionary ($\cdot_{\text{wik.}}$) and the union of these dictionaries ($\cdot_{\text{union}}$), respectively. The search spaces of the models trained with coupled constraints ($\widehat{\mathcal{Y}}\ldots$) are each pruned with the respective tag dictionary used to derive the coupled constraints. The observed difference between $\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{wik.}}$+C+L and $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L is statistically significant at $p < 0.01$ (**) and $p < 0.015$ (*) according to a paired bootstrap test (Efron and Tibshirani, 1993). Significance was not assessed for *avg* or *avg (8)*.

We next add monolingual cluster features to the model with the union dictionary. This model, $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C, significantly outperforms all other type-constrained models, demonstrating the utility of word-cluster features.[9] For further exploration, we train the same model on the datasets containing 500K tokens sampled from the target side of the parallel data ($\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L); this is done to explore the effects of large data during training. We find that training on these datasets result in an average accuracy of 87.2% which is comparable to the 87.3% reported for $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C in Table 1. This shows that the different source domain and amount of training data does not influence the performance of the HMM significantly.

Finally, we train CRF models where we treat type constraints as a partially observed lattice and use the full unpruned lattice for computing the partition function (§3.2). Due to space considerations, the results of these experiments are not shown in table 1. We observe similar trends in these results, but on average, accuracies are much lower compared to the type-constrained HMM models; the CRF model with the union dictionary along with cluster features achieves an average accuracy of 79.3% when trained on same data. This result is not unsurprising. First, the CRF's search space is fully unconstrained. Second, the dictionary only provides a weak set of observation constraints, which do not provide sufficient information to successfully train a discriminative model. However, as we will observe next, coupling the dictionary constraints with token-level information solves this problem.

4.3 Models with Token and Type Constraints

We now proceed to add token-level information, focusing in particular on coupled token and type

[9] These are monolingual clusters. Bilingual clusters as introduced in Täckström et al. (2012) might bring additional benefits.

constraints. Since it is not possible to generate projected token constraints for our monolingual treebanks, we train all models in this subsection on the 500K-tokens datasets sampled from the bitext. As a baseline, we first train HMM and CRF models that use *only* projected token constraints ($\tilde{y}^{\text{HMM}}$+C+L and $\tilde{y}^{\text{CRF}}$+C+L). As shown in Table 2, these models underperform the best type-level model ($\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L),[10] which confirms that projected token constraints are not reliable on their own. This is in line with similar projection models previously examined by Das and Petrov (2011).

We then study models with coupled token and type constraints. These models use the same three dictionaries as used in §4.2, but additionally couple the derived type constraints with projected token constraints; see the caption of Table 2 for a list of these models. Note that since we only allow projected tags that are licensed by the dictionary (Step 3 of the transfer, §2.3), the actual token constraints used in these models vary with the different dictionaries.

From Table 2, we see that coupled constraints are superior to token constraints, when used both with the HMM and the CRF. However, for the HMM, coupled constraints do not provide any benefit over type constraints alone, in particular when the projected dictionary or the union dictionary is used to derive the coupled constraints ($\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{proj.}}$+C+L and $\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{union}}$+C+L). We hypothesize that this is because these dictionaries (in particular the former) have the same bias as the token-level tag projections, so that the dictionary is unable to correct the systematic errors in the projections (see §2.1). Since the token constraints are stronger than the type constraints in the coupled models, this bias may have a substantial impact. With the Wiktionary dictionary, the difference between the type-constrained and the coupled-constrained HMM is negligible: $\mathcal{Y}^{\text{HMM}}_{\text{union}}$+C+L and $\widehat{\mathcal{Y}}^{\text{HMM}}_{\text{wik.}}$+C+L both average at an accuracy of 82.8%.

The CRF model, on the other hand, is able to take advantage of the complementary information in the coupled constraints, provided that the dictionary is able to filter out the systematic token-level errors. With a dictionary derived from Wiktionary and projected token-level constraints, $\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{wik.}}$+C+L performs better than all the remaining models, with an average accuracy of 88.8% across the eight Indo-European languages available to D&P and LG&T. Averaged over all 15 languages, its accuracy is 84.5%.

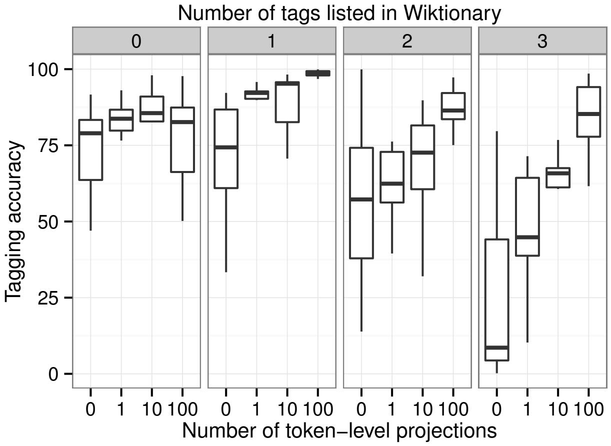

Figure 4: Relative influence of token and type constraints on tagging accuracy in the $\widehat{\mathcal{Y}}^{\text{CRF}}_{\text{wik.}}$+C+L model. Word types are categorized according to a) their number of Wiktionary tags (0,1,2 or 3+ tags, with 0 representing no Wiktionary entry; top-axis) and b) the number of times they are token-constrained in the training set (divided into buckets of 0, 1-9, 10-99 and 100+ occurrences; x-axis). The boxes summarize the accuracy distributions across languages for each word type category as defined by a) and b). The horizontal line in each box marks the median accuracy, the top and bottom mark the first and third quantile, respectively, while the whiskers mark the minimum and maximum values of the accuracy distribution.

5 Further Analysis

In this section we provide a detailed analysis of the impact of token versus type constraints and we study the pruning and filtering mistakes resulting from incomplete Wiktionary entries in detail. This analysis is based on the training portion of each treebank.

5.1 Influence of Token and Type Constraints

The empirical success of the model trained with coupled token and type constraints confirms that these constraints indeed provide complementary signals. Figure 4 provides a more detailed view of the relative benefits of each type of constraint. We observe several interesting trends.

First, word types that occur with more token constraints during training are generally tagged more accurately, regardless of whether these types occur

[10] To make the comparison fair vis-a-vis potential divergences in training domains, we compare to the best type-constrained model trained on the same 500K tokens training sets.

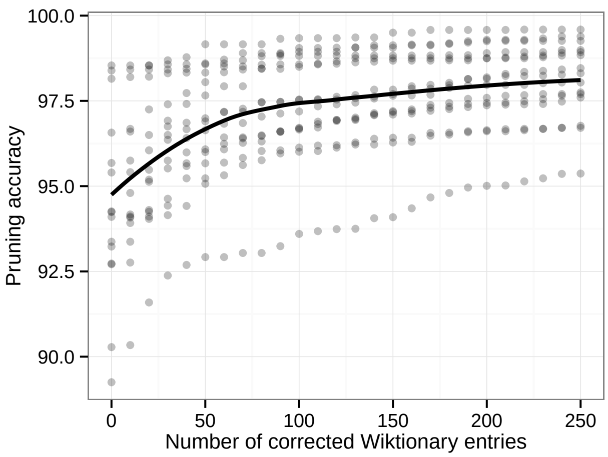

Figure 5: Average pruning accuracy (line) across languages (dots) as a function of the number of hypothetically corrected Wiktionary entries for the k most frequent word types. For example, position 100 on the x-axis corresponds to manually correcting the entries for the 100 most frequent types, while position 0 corresponds to experimental conditions.

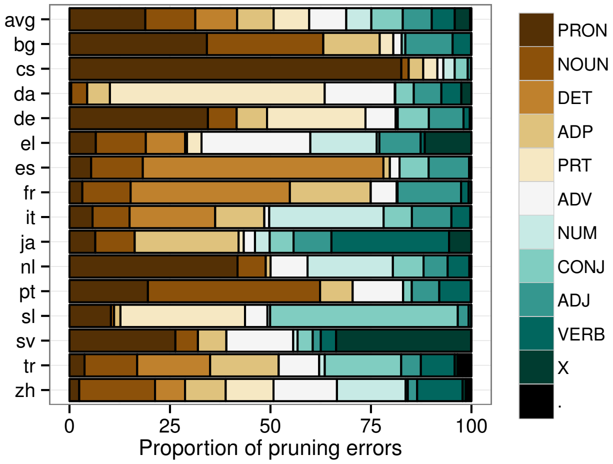

Figure 6: Prevalence of pruning mistakes per POS tag, when pruning the inference search space with Wiktionary.

in Wiktionary. The most common scenario is for a word type to have exactly one tag in Wiktionary and to occur with this projected tag over 100 times in the training set (facet 1, rightmost box). These common word types are typically tagged very accurately across all languages.

Second, the word types that are ambiguous according to Wiktionary (facets 2 and 3) are predominantly frequent ones. The accuracy is typically lower for these words compared to the unambiguous words. However, as the number of projected token constraints is increased from zero to 100+ observations, the ambiguous words are effectively disambiguated by the token constraints. This shows the advantage of intersecting token and type constraints.

Finally, projection generally helps for words that are not in Wiktionary, although the accuracy for these words never reach the accuracy of the words with only one tag in Wiktionary. Interestingly, words that occur with a projected tag constraint less than 100 times are tagged more accurately for types not in the dictionary compared to ambiguous word types with the same number of projected constraints. A possible explanation for this is that the ambiguous words are inherently more difficult to predict and that most of the words that are not in Wiktionary are less common words that tend to also be less ambiguous.

5.2 Wiktionary Pruning Mistakes

The error analysis by Li et al. (2012) showed that the tags licensed by Wiktionary are often valid. When using Wiktionary to prune the search space of our constrained models and to filter token-level projections, it is also important that correct tags are not mistakenly pruned because they are missing from Wiktionary. While the accuracy of filtering is more difficult to study, due to the lack of a gold standard tagging of the bitext, Figure 5 (position 0 on the x-axis) shows that search space pruning errors are not a major issue for most languages; on average the pruning accuracy is almost 95%. However, for some languages such as Chinese and Czech the correct tag is pruned from the search space for nearly 10% of all tokens. When using Wiktionary as a pruner, the upper bound on accuracy for these languages is therefore only around 90%. However, Figure 5 also shows that with some manual effort we might be able to remedy many of these errors. For example, by adding missing valid tags to the 250 most common word types in the worst language, the minimum pruning accuracy would rise above 95% from below 90%. If the same was to be done for all of the studied languages, the mean pruning accuracy would reach over 97%.

Figure 6 breaks down pruning errors resulting from incorrect or incomplete Wiktionary entries across the correct POS tags. From this we observe that, for many languages, the pruning errors are highly skewed towards specific tags. For example, for Czech over 80% of the pruning errors are caused by mistakenly pruned pronouns.

6 Conclusions

We considered the problem of constructing multilingual POS taggers for resource-poor languages. To this end, we explored a number of different models that combine token constraints with type constraints from different sources. The best results were obtained with a partially observed CRF model that effectively integrates these complementary constraints. In an extensive empirical study, we showed that this approach substantially improves on the state of the art in this context. Our best model significantly outperformed the second-best model on 10 out of 15 evaluated languages, when trained on identical data sets, with an insignificant difference on 3 languages. Compared to the prior state of the art (Li et al., 2012), we observed a relative reduction in error by 25%, averaged over the eight languages common to our studies.

Acknowledgments

We thank Alexander Rush for help with the hypergraph framework that was used to implement our models and Klaus Macherey for help with the bitext extraction. This work benefited from many discussions with Yoav Goldberg, Keith Hall, Kuzman Ganchev and Hao Zhang. We also thank the editor and the three anonymous reviewers for their valuable feedback. The first author is grateful for the financial support from the Swedish National Graduate School of Language Technology (GSLT).

References

Anne Abeillé, Lionel Clément, and François Toussenel. 2003. Building a Treebank for French. In A. Abeillé, editor, *Treebanks: Building and Using Parsed Corpora*, chapter 10. Kluwer.

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceedings of NAACL-HLT*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of CoNLL*.

Stanley F Chen. 2003. Conditional and joint models for grapheme-to-phoneme conversion. In *Proceedings of Eurospeech*.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised POS induction: How far have we come? In *Proceedings of EMNLP*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of ACL-HLT*.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39.

John DeNero and Klaus Macherey. 2011. Model-based aligner combination using dual decomposition. In *Proceedings of ACL-HLT*.

Brad Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York, NY, USA.

Victoria Fossum and Steven Abney. 2005. Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In *Proceedings of IJCNLP*.

Dan Garrette and Jason Baldridge. 2012. Type-supervised hidden markov models for part-of-speech tagging with incomplete tag dictionaries. In *Proceedings of EMNLP-CoNLL*.

Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008. EM can find pretty good HMM POS-taggers (when given a good start). In *Proceedings of ACL-HLT*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT Summit*.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Shen Li, João Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of EMNLP-CoNLL*.

Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19(2).

Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. 2009. Multilingual part-of-speech tagging: Two unsupervised approaches. *JAIR*, 36.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of EMNLP-CoNLL*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of LREC*.

Sujith Ravi and Kevin Knight. 2009. Minimized models for unsupervised part-of-speech tagging. In *Proceedings of ACL-IJCNLP*.

Stefan Riezler, Tracy H. King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell, III, and Mark Johnson. 2002. Parsing the wall street journal using a lexical-functional grammar and discriminative estimation techniques. In *Proceedings of ACL*.

Noah Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of ACL*.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of NAACL-HLT*.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of ACL*.

UN. 2006. ODS UN parallel corpus.

Jakob Uszkoreit and Thorsten Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *Proceedings of ACL-HLT*.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of COLING*.

Chenhai Xi and Rebecca Hwa. 2005. A backoff model for bootstrapping resources for non-English languages. In *Proceedings of HLT-EMNLP*.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of NAACL*.