What Can We Get From 1000 Tokens?
A Case Study of Multilingual POS Tagging For Resource-Poor Languages

Long Duong,[12] **Trevor Cohn,**[1] **Karin Verspoor,**[1] **Steven Bird,**[1] and **Paul Cook**[1]
[1]Department of Computing and Information Systems,
The University of Melbourne
[2]National ICT Australia, Victoria Research Laboratory
lduong@student.unimelb.edu.au
{t.cohn, karin.verspoor, sbird, paulcook}@unimelb.edu.au

Abstract

In this paper we address the problem of multilingual part-of-speech tagging for resource-poor languages. We use parallel data to transfer part-of-speech information from resource-rich to resource-poor languages. Additionally, we use a small amount of annotated data to learn to "correct" errors from projected approach such as tagset mismatch between languages, achieving state-of-the-art performance (91.3%) across 8 languages. Our approach is based on modest data requirements, and uses minimum divergence classification. For situations where no universal tagset mapping is available, we propose an alternate method, resulting in state-of-the-art 85.6% accuracy on the resource-poor language Malagasy.

1 Introduction

Part-of-speech (POS) tagging is a crucial task for natural language processing (NLP) tasks, providing basic information about syntax. Supervised POS tagging has achieved great success, reaching as high as 95% accuracy for many languages (Petrov et al., 2012). However, supervised techniques need manually annotated data, and this is either lacking or limited in most resource-poor languages. Fully unsupervised POS tagging is not yet useful in practice due to low accuracy (Christodoulopoulos et al., 2010). In this paper, we propose a semi-supervised method to narrow the gap between supervised and unsupervised approaches. We demonstrate that even a small amount of supervised data leads to substantial improvement.

Our method is motivated by the availability of parallel data. Thanks to the development of multilingual documents from government projects, book translations, multilingual websites, and so forth, parallel data between resource-rich and resource-poor languages is relatively easy to acquire. This parallel data provides the bridge that permits us to transfer POS information from a resource-rich to a resource-poor language.

Systems that make use of cross-lingual tag projection typically face several issues, including mismatches between the tagsets used for the languages, artifacts from noisy alignments and cross-lingual syntactic divergence. Our approach compensates for these issues by training on a small amount of annotated data on the target side, demonstrating that only 1k tokens of annotated data is sufficient to improve performance.

We first tag the resource-rich language using a supervised POS tagger. We then project POS tags from the resource-rich language to the resource-poor language using parallel word alignments. The projected labels are noisy, and so we use various heuristics to select only "good" training examples. We train the model in two stages. First, we build a maximum entropy classifier T on the (noisy) projected data. Next, we train a supervised classifier P on a small amount of annotated data (1,000 tokens) in the target language, using a minimum divergence technique to incorporate the first model, T. Compared with the state of the art (Täckström et al., 2013), we make more-realistic assumptions (e.g. relying on a tiny amount of annotated data rather than a huge crowd-sourced dictionary) and use less parallel data, yet achieve a better overall result. We achieved 91.3% average accuracy over 8 languages, exceeding Täckström et al. (2013)'s result of 88.8%.

The test data we employ makes use of mappings from language-specific POS tag inventories to a universal tagset (Petrov et al., 2012). However, such a mapping might not be available for resource-poor languages. Therefore, we also pro-

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 886–897,
October 25-29, 2014, Doha, Qatar. ©2014 Association for Computational Linguistics

pose a variant of our method which removes the need for identical tagsets between the projection model T and the correction model P, based on a two-output maximum entropy model over tag pairs. Evaluating on the resource-poor language Malagasy, we achieved 85.6% accuracy, exceeding the state-of-the-art of 81.2% (Garrette et al., 2013).

2 Background and Related Work

There is a wealth of prior work on multilingual POS tagging. The simplest approach takes advantage of the typological similarities that exist between languages pairs such as Czech and Russian, or Serbian and Croatian. They build the tagger — or estimate part of the tagger — on one language and apply it to the other language (Reddy and Sharoff, 2011, Hana et al., 2004).

Yarowsky and Ngai (2001) pioneered the use of parallel data for projecting tag information from a resource-rich language to a resource-poor language. Duong et al. (2013b) used a similar method on using sentence alignment scores to rank the goodness of sentences. They trained a seed model from a small part of the data, then applied this model to the rest of the data using self-training with revision.

Das and Petrov (2011) also used parallel data but additionally exploited graph-based label propagation to expand the coverage of labelled tokens. Each node in the graph represents a trigram in the target language. Each edge connects two nodes which have similar context. Originally, only some nodes received a label from direct label projection, and then labels were propagated to the rest of the graph. They only extracted the dictionary from the graph because the labels of nodes are noisy. They used the dictionary as the constraints for a feature-based HMM tagger (Berg-Kirkpatrick et al., 2010). Both Duong et al. (2013b) and Das and Petrov (2011) achieved 83.4% accuracy on the test set of 8 European languages.

Goldberg et al. (2008) pointed out that, with the presence of a dictionary, even an incomplete one, a modest POS tagger can be built using simple methods such as expectation maximization. This is because most of the time, words have a very limited number of possible tags, thus a dictionary that specifies the allowable tags for a word helps to restrict the search space. With a gold-standard dictionary, Das and Petrov (2011) achieved an accuracy of approximately 94% on the same 8 languages. The effectiveness of a gold-standard dictionary is undeniable, however it is costly to build one, especially for resource-poor languages. Li et al. (2012) used the dictionary from Wiktionary,[1] a crowd-sourced dictionary. They scored 84.8% accuracy on the same 8 languages. Currently, Wiktionary covers over 170 languages, but the coverage varies substantially between languages and, unsurprisingly, it is poor for resource-poor languages. Therefore, relying on Wiktionary is not effective for building POS taggers for resource-poor languages.

Täckström et al. (2013) combined both token information (from direct projected data) and type constraints (from Wiktionary's dictionary) to form the state-of-the-art multilingual tagger. They built a tag lattice and used these token and type constraints to prune it. The remaining paths are the training data for a CRF tagger. They achieved 88.8% accuracy on the same 8 languages.

Table 1 summarises the performance of the above models across all 8 languages. Note that these methods vary in their reliance on external resources. Duong et al. (2013b) use the least, i.e. only the Europarl Corpus (Koehn, 2005). Das and Petrov (2011) additionally use the United Nation Parallel Corpus. Li et al. (2012) didn't use any parallel text but used Wiktionary instead. Täckström et al. (2013) exploited more parallel data than Das and Petrov (2011) and also used a dictionary from Li et al. (2012).

Another approach for resource-poor languages is based on the availability of a small amount of annotated data. Garrette et al. (2013) built a POS tagger for Kinyarwanda and Malagasy. They didn't use parallel data but instead exploited four hours of manual annotation to build ∼4,000 tokens or ∼3,000 word-types of annotated data. These tokens or word-types were used to build a tag dictionary. They employed label propagation for expanding the coverage of this dictionary in a similar vein to Das and Petrov (2011), but they also used an external dictionary. They built training examples using the combined dictionary and then trained the tagger on this data. They achieved 81.9% and 81.2% accuracy for Kinyarwanda and Malagasy respectively. Note that their usage of an external dictionary compromises their claim of using only 4 hours of annotation.

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

	da	nl	de	el	it	pt	es	sv	Average
Das and Petrov (2011)	83.2	79.5	82.8	82.5	86.8	87.9	84.2	80.5	83.4
Duong et al. (2013b)	85.6	84.0	85.4	80.4	81.4	86.3	83.3	81.0	83.4
Li et al. (2012)	83.3	86.3	85.4	79.2	86.5	84.5	86.4	86.1	84.8
Täckström et al. (2013)	88.2	85.9	90.5	89.5	89.3	91.0	87.1	88.9	88.8

Table 1: Previously published token-level POS tagging accuracy for various models across 8 languages — Danish (da), Dutch (nl), German (de), Greek (el), Italian (it), Portuguese (pt), Spanish (es), Swedish (sv) — evaluated on CoNLL data (Buchholz and Marsi, 2006).

The method we propose in this paper is similar in only using a small amount of annotation. However, we directly use the annotated data to train the model rather than using a dictionary. We argue that with a proper "guide", we can take advantage of very limited annotated data.

2.1 Annotated data

Our annotated data mainly comes from CoNLL shared tasks on dependency parsing (Buchholz and Marsi, 2006). The language specific tagsets are mapped into the universal tagset. We will use this annotated data mainly for evaluation. Table 2 shows the size of annotated data for each language. The 8 languages we are considering in this experiment are not actually resource-poor languages. However, running on these 8 languages makes our system comparable with previously proposed methods. Nevertheless, we try to use as few resources as possible, in order to simulate the situation for resource-poor languages. Later in Section 6 we adapt the approach for Malagasy, a truly resource-poor language.

2.2 Universal tagset

We employ the universal tagset from (Petrov et al., 2012) for our experiment. It consists of 12 common tags: *NOUN, VERB, ADJ* (adjective), *ADV* (adverb), *PRON* (pronoun), *DET* (determiner and article), *ADP* (preposition and postposition), *CONJ* (conjunctions), *NUM* (numerical), *PRT* (particle), *PUNC* (punctuation) and *X* (all other categories including foreign words and abbreviations). Petrov et al. (2012) provide the mapping from each language-specific tagset to the universal tagset.

The idea of using the universal tagset is of great use in multilingual applications, enabling comparison across languages. However, the mapping is not always straightforward. Table 2 shows the size of the annotated data for each language, the number of tags presented in the data, and the list of tags that are not matched. We can see that only 8 tags are presented in the annotated data for Danish, i.e, 4 tags (*DET, PRT, PUNC,* and *NUM*) are missing.[2] Thus, a classifier using all 12 tags will be heavily penalized in the evaluation.

Li et al. (2012) considered this problem and tried to manually modify the Danish mappings. Moreover, *PRT* is not really a universal tag since it only appears in 3 out of the 8 languages. Plank et al. (2014) pointed out that *PRT* often gets confused with *ADP* even in English. We will later show that the mapping problem causes substantial degradation in the performance of a POS tagger exploiting parallel data. The method we present here is more target-language oriented: our model is trained on the target language, in this way, only relevant information from the source language is retained. Thus, we automatically correct the mapping, and other incompatibilities arising from incorrect alignments and syntactic divergence between the source and target languages.

Lang	Size(k)	# Tags	Not Matched
da	94	8	DET, PRT, PUNC, NUM
nl	203	11	PRT
de	712	12	
el	70	12	
it	76	11	PRT
pt	207	11	PRT
es	89	11	PRT
sv	191	11	DET
AVG	205		

Table 2: The size of annotated data from CoNLL (Buchholz and Marsi, 2006), and the number of tags included and missing for 8 languages.

[2] Many of these are mistakes in the mapping, however, they are indicative of the kinds of issues expected in low-resource languages.

3 Directly Projected Model (DPM)

In this section we describe a maximum entropy tagger that only uses information from directly projected data.

3.1 Parallel data

We first collect Europarl data having English as the source language, an average of 1.85 million parallel sentences for each of the 8 language pairs. In terms of parallel data, we use far less data compared with other recent work. Das and Petrov (2011) used Europarl and the ODS United Nation dataset, while Täckström et al. (2013) additionally used parallel data crawled from the web. The amount of parallel data is crucial for alignment quality. Since DPM uses alignments to transfer tags from source to target language, the performance of DPM (and other models that exploit projection) largely depends on the quantity of parallel data. The "No LP" model of Das and Petrov (2011), which only uses directly projected labels (without label propagation), scored 81.3% for 8 languages. However, using the same model but with more parallel data, Täckström et al. (2013) scored 84.9% on the same test set.

3.2 Label projection

We use the standard alignment tool Giza++ (Och and Ney, 2003) to word align the parallel data. We employ the Stanford POS tagger (Toutanova et al., 2003) to tag the English side of the parallel data and then project the label to the target side. It has been confirmed in many studies (Täckström et al., 2013, Das and Petrov, 2011, Toutanova and Johnson, 2008) that directly projected labels are noisy. Thus we need a method to reduce the noise. We employ the strategy of Yarowsky and Ngai (2001) of ranking sentences using a their alignment scores from IBM model 3.

Firstly, we want to know how noisy the projected data is. Thus, we use the test data to build a simple supervised POS tagger using the TnT tagger (Brants, 2000) which employs a second-order Hidden Markov Model (HMM). We tag the projected data and compare the label from direct projection and from the TnT tagger. The labels from the TnT Tagger are considered as pseudo-gold labels. Column "Without Mapping" from Table 3 shows the average accuracy for the first n-sentences ($n = 60k, 100k, 200k, 500k$) for 8 languages according to the ranking. Column "Coverage" shows the percentages of projected label (the other tokens are Null aligned). We can see that when we select more data, both coverage and accuracy fall. In other words, using the sentence alignment score, we can rank sentences with high coverage and accuracy first. However, even after ranking, the accuracy of projected labels is less than 80% demonstrating how noisy the projected labels are.

Table 3 (column "With Mapping") additionally shows the accuracy using simple tagset mapping, i.e. mapping each tag to the tag it is assigned most frequently in the test data. For example *DET, PRT, PUNC, NUM*, missing from Danish gold data, will be matched to *PRON, X, X, ADJ* respectively. This simple matching yields a $\sim$ 4% (absolute) improvement in average accuracy. This illustrates the importance of handling tagset mapping carefully.

3.3 The model

In this section, we introduce a maximum entropy tagger exploiting the projected data. We select the first 200k sentences from Table 3 for this experiment. This number represents a trade-off between size and accuracy. More sentences provide more information but at the cost of noisier data. Duong et al. (2013b) also used sentence alignment scores to rank sentences. Their model stabilizes after using 200k sentences. We conclude that 200k sentences is enough and capture most information from the parallel data.

Features	Descriptions
W@-1	Previous word
W@+1	Next word
W@0	Current word
CAP	First character is capitalized
NUMBER	Is number
PUNCT	Is punctuation
SUFFIX@k	Suffix up to length 3 ($k <= 3$)
WC	Word class

Table 4: Feature template for a maximum entropy tagger

We ignore tokens that don't have labels, which arise from null alignments and constitute approximately 14% of the data. The remaining data ($\sim$1.4 million tokens) are used to train a maximum entropy (MaxEnt) model. MaxEnt is one of the simplest forms of probabilistic classifier, and is appropriate in this setting due to the incomplete

Data Size (k)	Coverage (%)	Without Mapping	With Mapping
60	91.5	79.9	84.2
100	89.1	79.4	83.6
200	86.1	79.1	82.9
500	82.4	78.0	81.5

Table 3: The coverage, and POS tagging accuracy with and without tagset mapping of directly projected labels, averaged over 8 languages for different data sizes

Model	da	nl	de	el	it	pt	es	sv	Avg
All features	64.4	83.3	86.3	79.7	82.0	86.5	82.5	76.5	80.2
- Word Class	64.7	82.6	86.6	79.0	82.8	84.6	82.2	76.9	79.9
- Suffix	64.0	82.8	86.3	78.1	81.0	85.9	82.3	76.2	79.6
- Prev, Next Word	62.6	82.5	87.4	79.0	81.9	86.5	82.2	74.8	79.6
- Cap, Num, Punct	64.0	81.9	84.0	78.0	79.1	86.3	81.8	75.6	78.8

Table 5: The accuracy of Directed Project Model (DPM) with different feature sets, removing one feature set at a time

sequence data. While sequence models such as HMMs or CRFs can provide more accurate models of label sequences, they impose a more stringent training requirement.[3] We also experimented with a first-order linear chain CRF trained on contiguous sub-sequences but observed $\sim 4\%$ (absolute) drop in performance.

The maximum entropy classifier estimates the probability of tag t given a word w as

$$P(t|w) = \frac{1}{Z(w)} \exp \sum_{j=1}^{D} \lambda_j f_j(w,t) \,,$$

where $Z(w) = \sum_t \exp \sum_{j=1}^{D} \lambda_j f_j(w,t)$ is the normalization factor to ensure the probabilities $P(t|w)$ sum to one. Here f_j is a feature function and λ_j is the weight for this feature, learned as part of training. We use Maximum A Posteriori (MAP) estimation to maximize the log likelihood of the training data, $\mathcal{D} = \{w_i, t_i\}_{i=1}^{N}$, subject to a zero-mean Gaussian regularisation term,

$$\begin{aligned}\mathcal{L} &= \log P(\Lambda) \prod_{i=1}^{N} P(t^{(i)}|w^{(i)}) \\ &= -\sum_{j=1}^{D} \frac{\lambda_j^2}{2\delta^2} + \sum_{i=1}^{N} \sum_{j=1}^{D} \lambda_j f_j(w_i,t_i) - \log Z(w_i)\end{aligned}$$

where the regularisation term limits over-fitting, an important concern when using large feature sets. For our experiments we set $\delta^2 = 1$. We use L-BFGS which performs gradient ascent to maximize $\mathcal{L}$. Table 4 shows the features we considered for building the DPM. We use $mkcls$, an unsupervised method for word class induction which is widely used in machine translation (Och, 1999). We run $mkcls$ to obtain 100 word classes, using only the target language side of the parallel data.

Table 5 shows the accuracy of the DPM evaluated on 8 languages ("All features model"). DPM performs poorly on Danish, probably because of the tagset mapping issue discussed above. The DPM result of 80.2% accuracy is encouraging, particularly because the model had no explicit supervision.

To see what features are meaningful for our model, we remove features in turn and report the result. The result in Table 5 disagrees with Täckström et al. (2013) on the word class features. They reported a gain of approximately 3% (absolute) using the word class. However, it seems to us that these features are not especially meaningful (at least in the present setting). Possible reasons for the discrepancy are that they train the word class model on a massive quantity of external monolingual data, or their algorithms for word clustering are better (Uszkoreit and Brants, 2008). We can see that the most informative features are Capitalization, Number and Punctuation. This makes sense because in languages such as German, capitalization is a strong indicator of *NOUN*. Number and punctuation features ensure that we classify *NUM* and *PUNCT* tags correctly.

[3] Täckström et al. (2013) train a CRF on incomplete data, using a tag dictionary heuristic to define a 'gold standard' lattice over label sequences.

4 Correction Model

In this section we incorporate the directly projected model into a second *correction* model trained on a small supervised sample of 1,000 annotated tokens. Our DPM model is not very accurate; as we have discussed it makes many errors, due to invalid or inconsistent tag mappings, noisy alignments, and cross-linguistic syntactic divergence. However, our aim is to see how effectively we can exploit the strengths of the DPM model while correcting for its inadequacies using direct supervision. We select only 1,000 annotated tokens to reflect a low resource scenario. A small supervised training sample is a more realistic form of supervision than a tag dictionary (noisy or otherwise). Although used in most prior work, a tag dictionary for a new language requires significant manual effort to construct. Garrette and Baldridge (2013) showed that a 1,000 token dataset could be collected very cheaply, requiring less than 2 hours of non-expert time.

Our correction model makes use of a *minimum divergence* (MD) model (Berger et al., 1996), a variant of the maximum entropy model which biases the target distribution to be similar to a static reference distribution. The method has been used in several language applications including machine translation (Foster, 2000) and parsing (Plank and van Noord, 2008, Johnson and Riezler, 2000). These previous approaches have used various sources of reference distribution, e.g., incorporating information from a simpler model (Johnson and Riezler, 2000) or combining in- and out-of-domain models (Plank and van Noord, 2008). Plank and van Noord (2008) concluded that this method for adding prior knowledge only works with high quality reference distributions, otherwise performance suffers.

In contrast to these previous approaches, we consider the specific setting where both the learned model and the reference model $s_o = P(t|w)$ are both maximum entropy models. In this case we show that the MD setup can be simplified to a regularization term, namely a Gaussian prior with a non-zero mean. We model the classification probability, $P'(t|w)$ as the product between a base model and a maximum entropy classifier,

$$P'(t|w) \propto P(t|w) \exp \sum_{j=1}^{D} \gamma_j f_j(w,t)$$

where here we use the DPM model as base model $P(t|w)$. Under this setup, where P' uses the same features as P, and both are log-linear models, this simplifies to

$$\begin{aligned} P'(t|w) &\propto \exp \left(\sum_{j=1}^{D} \lambda_j f_j(w,t) + \sum_{j=1}^{D} \gamma_j f_j(w,t) \right) \\ &\propto \exp \sum_{j=1}^{D} (\lambda_j + \gamma_j) f_j(w,t) \qquad (1) \end{aligned}$$

where the constant of proportionality is $Z'(w) = \sum_t \exp \sum_{j=1}^{D} (\lambda_j + \gamma_j) f_j(w,t)$. It is clear that Equation (1) also defines a maximum entropy classifier, with parameters $\alpha_j = \lambda_j + \gamma_j$, and consequently this might seem to be a pointless exercise. The utility of this approach arises from the prior: MAP training with a zero mean Gaussian prior over γ is equivalent to a Gaussian prior over the aggregate weights, $\alpha_j \sim \mathcal{N}(\lambda_j, \sigma^2)$. This prior enforces parameter sharing between the two models by penalising parameter divergence from the underlying DPM model λ. The resulting training objective is

$$\mathcal{L}^{\text{corr}} = \log P(\mathbf{t}|\mathbf{w}, \alpha) - \frac{1}{2\sigma^2} \sum_{j=1}^{D} (\alpha_j - \lambda_j)^2$$

which can be easily optimised using standard gradient-based methods, e.g., L-BFGS. The contribution of the regulariser is scaled by the constant $\frac{1}{2\sigma^2}$.

4.1 Regulariser sensitivity

Careful tuning of the regularisation term σ^2 is critical for the correction model, both to limit overfitting on the very small training sample of 1,000 tokens, and to control the extent of the influence of the DPM model over the correction model. A larger value of σ^2 lessens the reliance on the DPM and allows for more flexible modelling of the training set, while a small value of σ^2 forces the parameters to be close to the DPM estimates at the expense of data fit. We expect the best value to be somewhere between these extremes, and use line-search to find the optimal value for σ^2. For this purpose, we hold out 100 tokens from the 1,000 instance training set, for use as our development set for hyper-parameter selection.

From Figure 1, we can see that the model performs poorly on small values of σ^2. This is understandable because the small σ^2 makes the model

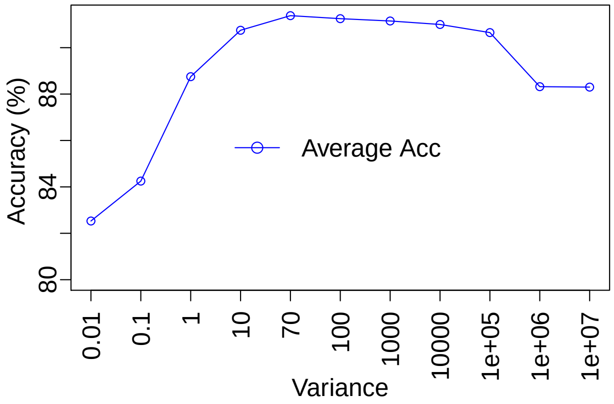

Figure 1: Sensitivity of regularisation parameter σ^2 against the average accuracy measured on 8 languages on the development set

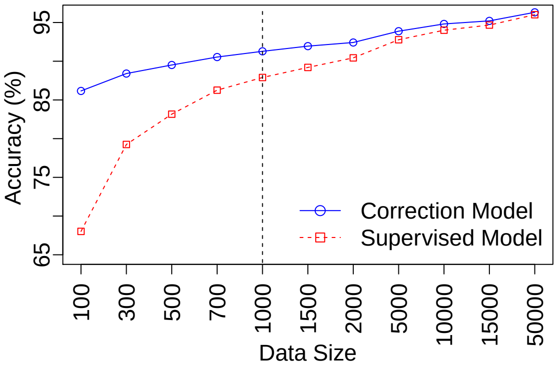

Figure 2: Learning curve for correction model and supervised model: the x-axis is the size of data (number of tokens); the y-axis is the average accuracy measured on 8 languages; the dashed line shows the data condition reported in Table 6

too similar to DPM, which is not very accurate (80.2%). At the other extreme, if σ^2 is large, the DPM model is ignored, and the correction model is equivalent with the supervised model ($\sim$ 88% accuracy). We select the value of $\sigma^2 = 70$, which maximizes the accuracy on the development set.

4.2 The model

Using the value of $\sigma^2 = 70$, we retrain the model on the whole 1,000-token training set and evaluate the model on the rest of the annotated data. Table 6 shows the performance of DPM, Supervised model, Correction model and the state-of-the-art model (Täckström et al., 2013). The supervised model trains a maximum entropy tagger using the same features as in Table 4 on this 1000 tokens. The only difference between the supervised model and the correction model is that in the correction model we additionally incorporate DPM as the prior.

The supervised model performs surprisingly well confirming that our features are meaningful in distinguishing between tags. This model achieves high accuracy on Danish compared with other languages probably because Danish is easier to learn since it contains only 8 tags. Despite the fact that the DPM is not very accurate, the correction model consistently outperforms the supervised model on all considered languages, approximately 4.3% (absolute) better on average. This shows that our method of incorporating DPM to the model is efficient and robust.

The correction model performs much better than the state-of-the-art for 7 languages but slightly worse for 1 language. On average we achieve 91.3% accuracy compared with 88.8% for the state-of-the-art, an error rate reduction of 22.3%. This is despite using fewer resources and only modest supervision.

5 Analysis

Tagset mismatch In the correction model, we implicitly resolve the mismatched tagset issue. DPM might contain tags that don't appear in the target language or generally are errors in the mapping. However, when incorporating DPM into the correction model, only the feature weight of tags that appear in the target language are retained. In general, because we don't explicitly do any mapping between languages, we might have trouble if the tagset size of the target language is bigger than the source language tagset. However, this is not the case for our experiment because we choose English as the source-side and English has the full 12 tags.

Learning curve We investigate the impact of the number of available annotated tokens on the correction model. Figure 2 shows the learning curve of the correction model and the supervised model. We can clearly see the differences between 2 models when the size of training data is small. For example, at 100 tokens, the difference is very large, approximately 18% (absolute), it is also 6% (absolute) better than DPM. This difference diminishes as we add more data. This make sense because when we add more data, the supervised model become stronger, while the effective-

Model	da	nl	de	el	it	pt	es	sv	Avg
DPM	64.4	83.3	86.3	79.7	82.0	86.5	82.5	76.5	80.2
Täckström et al. (2013)	88.2	85.9	90.5	89.5	89.3	91.0	87.1	**88.9**	88.8
Supervised model	90.1	84.6	89.6	88.2	81.4	87.6	88.9	85.4	87.0
Correction Model	**92.1**	**91.1**	**92.5**	**92.1**	**89.9**	**92.5**	**91.6**	88.7	**91.3**
DPM (with dict)	65.2	83.9	87.0	79.1	83.5	87.1	83.0	77.5	80.8
Correction Model (with dict)	93.3	92.2	93.7	93.2	92.2	93.1	92.8	90.0	92.6

Table 6: The comparison of our Directly Projected Model, Supervised Model, Correction Model and the state-of-the-art system (Täckström et al., 2013). The best performance for each language is shown in bold. The models that are built with a dictionary are provided for reference.

ness of the DPM prior on the correction model is wearing off. An interesting observation is that the correction model is always better, even when we add massive amounts of annotated data. At 50,000 tokens, when the supervised model reaches 96% accuracy, the correction model is still 0.3% (absolute) better, reaching 96.3%. It means that even at that high level of confidence, some information can still be added from DPM to the correction model. This improvement probably comes from the observation that the ambiguity in one language is explained through the alignment. It also suggests that this method could improve the performance of a supervised POS tagger even for resource-rich languages.

Our methods are also relevant for annotation projects for resource-poor languages. Assuming that it is very costly to annotate even 100 tokens, applying our methods can save annotation effort but maintain high performance. For example, we just need 100 tokens to match the accuracy of a supervised method trained on 700 tokens, or we just need 500 tokens to match the performance with nearly 2,000 tokens of supervised learning.

Our method is simple, but particularly suitable for resource-poor languages. We need a small amount of annotated data for a high performance POS tagger. For example, we need only around 300 annotated tokens to reach the same accuracy as the state-of-the-art unsupervised POS tagger (88.8%).

Tag dictionary Although, it is not our objective to rely on the dictionary, we are interested in whether the gains from the correction model still persist when the DPM performance is improved. We attempt to improve DPM, following the method of Li et al. (2012) by building a tag dictionary using Wiktionary. This dictionary is then used as a feature which fires for word-tag pairings present in the dictionary. We expect that when we add this additional supervision, the DPM model should perform better. Table 6 shows the performance of DPM and the correction model when incorporating the dictionary. The DPM model only increases 0.6% absolute but the correction model increases 1.3%. Additionally, it shows that our model can improve further by incorporating external information where available.

CRF Our approach of using simple classifiers begs the question of whether better results could be obtained using sequence models, such as conditional random fields (CRFs). As mentioned previously, a CRF is not well suited for incomplete data. However, as our second 'correction' model is trained on complete sequences, we now consider using a CRF in this stage. The training algorithm is as follows: first we estimate the DPM feature weights on the incomplete data as before, and next we incorporate the feature weights into a CRF trained on the 1,000 annotated tokens. This is complicated by the different feature sets between the MaxEnt classifier and the CRF, however the classifier uses a strict subset of the CRF features. Thus, we use the minimum divergence prior for the token level features, and a standard zero-mean prior for the sequence features. That is, the objective function of the CRF correction model becomes:

$$\begin{aligned}\mathcal{L}_{\text{crf}}^{\text{corr}} &= \log P(\mathbf{t}|\mathbf{w}, \alpha) \\ &- \frac{1}{2\delta_1^2}\sum_{j\in F_1}(\alpha_j - \lambda_j)^2 - \frac{1}{2\delta_2^2}\sum_{j\in F_2}\alpha_j^2 \quad (2)\end{aligned}$$

where F_1 is the set of features referring to only one label as in the DPM maxent model and F_2 is the set of features over label pairs. The union of $F = F_1 \cup F_2$ is the set of all features for the CRF. We perform grid search using held out data as before for δ_1^2 and δ_2^2. The CRF correction model scores 88.1% compared with 86.5% of the supervised CRF model trained on the 1,000 tokens. Clearly, this is beneficial, however, the CRF correction model still performs worse than the MaxEnt correction model (91.3%). We are not sure why but one reason might be overfitting of the CRF, due to its large feature set and tiny training sample. Moreover, this CRF approach is orthogonal to Täckström et al. (2013): we could use their CRF model as the DPM model and train the CRF correction model using the same minimum divergence method, presumably resulting in even higher performance.

6 Two-output model

Garrette and Baldridge (2013) also use only a small amount of annotated data, evaluating on two resource-poor languages Kinyarwanda (KIN) and Malagasy (MLG). As a simple baseline, we trained a maxent supervised classifier on this data, achieving competitive results of 76.4% and 80.0% accuracy compared with their published results of 81.9% and 81.2% for KIN and MLG, respectively. Note that the Garrette and Baldridge (2013) method is much more complicated than this baseline, and additionally uses an external dictionary.

We want to further improve the accuracy of MLG using parallel data. Applying the technique from Section 4 will not work directly, due to the tagset mismatch (the Malagasy tagset contains 24 tags) which results in highly different feature sets. Moreover, we don't have the language expertise to manually map the tagset. Thus, in this section, we propose a method capable of handling tagset mismatch. For data, we use a parallel English-Malagasy corpus of ∼100k sentences,[4] and the POS annotated dataset developed by Garrette and Baldridge (2013), which comprises 4230 tokens for training and 5300 tokens for testing.

6.1 The model

Traditionally, MaxEnt classifiers are trained using a single label.[5] The method we propose is trained with pairs of output labels: one for the Malagasy tag (t_M) and one for the universal tag (t_U), which are both predicted conditioned on a Malagasy word (w_M) in context. Our two-output model is defined as

$$P(t_M, t_U|w_M) = \frac{1}{Z(w_M)} \exp\left(\sum_{j=1}^{D} \lambda_j f_j^M(w, t_M) + \sum_{j=1}^{E} \gamma_j f_j^U(w, t_U) + \sum_{j=1}^{F} \alpha_j f_j^B(w, t_M, t_U)\right) \tag{3}$$

where f^M, f^U, f^B are the feature functions considering t_M only, t_U only, and over both outputs t_M and t_U respectively, and $Z(w_M)$ is the partition function. We can think of Eq. (3) as the combination of 3 models: the Malagasy maxent supervised model, the DPM model, and the tagset mapping model. The central idea behind this model is to learn to predict not just the MLG tags, as in a standard supervised model, but also to learn the mapping between MLG and the noisy projected universal tags. Framing this as a two output model allows for information to flow both ways, such that confident taggings in either space can inform the other, and accordingly the mapping weights α are optimised to maximally exploit this effect.

One important question is how to obtain labelled data for training the two-output model, as our small supervised sample of MLG text is only annotated for MLG labels t_M. We resolve this by first learning the DPM model on the projected labels, after which we automatically label our correction training set with predicted tags from the DPM model. That is, we augment the annotated training data from (t_M, w_M) to become (t_M, t_U, w_M). This is then used to train the two-output maxent classifier, optimising a MAP objective using standard gradient descent. Note that it would be possible to apply the same minimum divergence technique for the two-output maxent model. In this case the correction model would include a regularization term over the λ to bias towards the DPM parameters, while γ and α would use a zero-mean regularizer. However, we leave this for future work.

Table 7 summarises the performance of the state-of-the-art (Garrette et al., 2013), the supervised model and the two-output maxent model evaluated on the Malagasy test set. The two-output maxent model performs much better than the supervised model, achieving ∼5.3% (absolute) im-

[4] http://www.ark.cs.cmu.edu/global-voices/

[5] Or else a sequence of labels, in the case of a conditional random field (Lafferty et al., 2001). However, even in this case, each token is usually assigned a single label. An exception is the factorial CRF (Sutton et al., 2007), which models several co-dependent sequences. Our approach is equivalent to a factorial CRF without edges between tags for adjacent tokens in the input.

Model	Accuracy (%)
Garrette et al. (2013)	81.2
MaxEnt Supervised	80.0
2-output MaxEnt (Universal tagset)	85.3
2-output MaxEnt (Penn tagset)	85.6

Table 7: The performance of different models for Malagasy.

provement. An interesting property of this approach is that we can use different tagsets for the DPM. We also tried the original Penn treebank tagset which is much larger than the universal tagset (48 vs. 12 tags). We observed a small improvement reaching 85.6%, suggesting that some pertinent information is lost in the universal tagset. All in all, this is a substantial improvement over the state-of-the-art result of 81.2% (Garrette et al., 2013) and an error reduction of 23.4%.

7 Conclusion

In this paper, we thoroughly review the work on multilingual POS tagging of the past decade. We propose a simple method for building a POS tagger for resource-poor languages by taking advantage of parallel data and a small amount of annotated data. Our method also efficiently resolves the tagset mismatch issue identified for some language pairs. We carefully choose and tune the model. Comparing with the state-of-the-art, we are using the more realistic assumption that a small amount of labelled data can be made available rather than requiring a crowd-sourced dictionary. We use less parallel data which as we pointed out in section 3.1, could have been a huge disadvantage for us. Moreover, we did not exploit any external monolingual data. Importantly, our method is simpler but performs better than previously proposed methods. With only 1,000 annotated tokens, less than 1% of the test data, we can achieve an average accuracy of 91.3% compared with 88.8% of the state-of-the-art (error reduction rate ∼22%). Across the 8 languages we are substantially better at 7 and slightly worse at one. Our method is reliable and could even be used to improve the performance of a supervised POS tagger.

Currently, we are building the tagger and evaluating through several layers of mapping. Each layer might introduce some noise which accumulates and leads to a biased model. Moreover, the tagset mappings are not available for many resource-poor languages. We therefore also proposed a method to automatically match between tagsets based on a two-output maximum entropy model. On the resource-poor language Malagasy, we achieved the accuracy of 85.6% compared with the state-of-the-art of 81.2% (Garrette et al., 2013). Unlike their method, we didn't use an external dictionary but instead use a small amount of parallel data.

In future work, we would like to improve the performance of DPM by collecting more parallel data. Duong et al. (2013a) pointed out that using a different source language can greatly alter the performance of the target language POS tagger. We would like to experiment with different source languages other than English. We assume that we have 1,000 tokens for each language. Thus, for the 8 languages we considered we will have 8,000 annotated tokens. Currently, we treat each language independently, however, it might also be interesting to find some way to incorporate information from multiple languages simultaneously to build the tagger for a single target language.

Acknowledgments

We would like to thank Dan Garreette, Jason Baldridge and Noah Smith for Malagasy and Kinyarwanda datasets. This work was supported by the University of Melbourne and National ICT Australia (NICTA). NICTA is funded by the Australian Federal and Victoria State Governments, and the Australian Research Council through the ICT Centre of Excellence program. Dr Cohn is the recipient of an Australian Research Council Future Fellowship (project number FT130101105).

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Proceeding of HLT-NAACL*, pages 582–590.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *COMPUTATIONAL LINGUISTICS*, 22:39–71.

Thorsten Brants. 2000. TnT: A statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing (ANLP '00)*, pages 224–231, Seattle, Washington, USA.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised pos induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 575–584.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 600–609.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013a. Increasing the quality and quantity of source language data for Unsupervised Cross-Lingual POS tagging. Proceedings of the Sixth International Joint Conference on Natural Language Processing, pages 1243–1249. Asian Federation of Natural Language Processing.

Long Duong, Paul Cook, Steven Bird, and Pavel Pecina. 2013b. Simpler unsupervised POS tagging with bilingual projections. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 634–639. Association for Computational Linguistics.

George Foster. 2000. A maximum entropy/minimum divergence translation model. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 45–52.

Dan Garrette and Jason Baldridge. 2013. Learning a part-of-speech tagger from two hours of annotation. pages 138–147, June.

Dan Garrette, Jason Mielens, and Jason Baldridge. 2013. Real-world semi-supervised learning of pos-taggers for low-resource languages. pages 583–592, August.

Yoav Goldberg, Meni Adler, and Michael Elhadad. 2008. Em can find pretty good hmm pos-taggers (when given a good start. In *In Proc. ACL*, pages 746–754.

Jiri Hana, Anna Feldman, and Chris Brew. 2004. A resource-light approach to Russian morphology: Tagging Russian using Czech resources. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP '04)*, pages 222–229, Barcelona, Spain, July.

Mark Johnson and Stefan Riezler. 2000. Exploiting auxiliary distributions in stochastic unification-based grammars. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 154–161.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, pages 79–86, Phuket, Thailand. AAMT.

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 the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289.

Shen Li, João V. Graça, and Ben Taskar. 2012. Wiki-ly supervised part-of-speech tagging. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1389–1398.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29(1):19–51, March.

Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics*, EACL '99, pages 71–76.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may. European Language Resources Association (ELRA).

Barbara Plank and Gertjan van Noord. 2008. Exploring an auxiliary distribution based approach to domain adaptation of a syntactic disambiguation model. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 9–16.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for*

Computational Linguistics, pages 742–751, Gothenburg, Sweden, April.

Siva Reddy and Serge Sharoff. 2011. Cross language POS taggers (and other tools) for Indian languages: An experiment with Kannada using Telugu resources. In *Proceedings of IJCNLP workshop on Cross Lingual Information Access: Computational Linguistics and the Information Need of Multilingual Societies. (CLIA 2011 at IJNCLP 2011)*, Chiang Mai, Thailand, November.

Charles Sutton, Andrew McCallum, and Khashayar Rohanimanesh. 2007. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *J. Mach. Learn. Res.*, 8:693–723, May.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Kristina Toutanova and Mark Johnson. 2008. A bayesian lda-based model for semi-supervised part-of-speech tagging. In J.C. Platt, D. Koller, and Y. Singer a nd S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1521–1528. Curran Associates, Inc.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1 (NAACL '03)*, pages 173–180, Edmonton, Canada.

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

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics on Language technologies*, NAACL '01, pages 1–8.