If You Even Don't Have a Bit of Bible: Learning Delexicalized POS Taggers

Zhiwei Yu,[†] **David Mareček,**[⋆] **Zdeněk Žabokrtský,**[⋆] **Daniel Zeman**[⋆]
[†]Department of Computer Science and Engineering, Shanghai Jiaotong University
[⋆]Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics, Charles University in Prague
jordanyzw@sjtu.edu.cn, {marecek,zabokrtsky,zeman}@ufal.mff.cuni.cz

Abstract

Part-of-speech (POS) induction is one of the most popular tasks in research on unsupervised NLP. Various unsupervised and semi-supervised methods have been proposed to tag an unseen language. However, many of them require some partial understanding of the target language because they rely on dictionaries or parallel corpora such as the Bible. In this paper, we propose a different method named *delexicalized tagging*, for which we only need a raw corpus of the target language. We transfer tagging models trained on annotated corpora of one or more resource-rich languages. We employ language-independent features such as word length, frequency, neighborhood entropy, character classes (alphabetic vs. numeric vs. punctuation) etc. We demonstrate that such features can, to certain extent, serve as predictors of the part of speech, represented by the universal POS tag (Das and Petrov, 2011).

Keywords: delexicalized tagging, HamdleDT 2.0, features expansion, classifier

1. Introduction

Part-of-speech (POS) tagging is sometimes considered an almost solved problem in NLP. Standard supervised approaches often reach accuracy above 95% if sufficiently large hand-labeled training data are available (typically several hundred thousand tokens or more). However, we still believe that it makes sense to study semi-supervised and unsupervised approaches because of the following reasons:

- It is hardly realistic to expect that manual annotation efforts will be ever invested into all 7,000 languages.
- Even if it might be very efficient to start at least with some small annotated data, we believe that adding new features independent of hand-tagged text might be helpful in a combination of supervised and unsupervised methods, e.g., for better handling of out-of-vocabulary words.
- We should keep in mind that the "standard" POS distinctions—although broadly used—are not manifested in languages directly. They result from certain linguistic tradition whose current dominance can be attributed to geopolitical reasons rather than its linguistic "obviousness". Thus, for instance, if we say that something is an adverb in language X, we should be able to support such a claim by some measurable evidence rather than just by saying that it becomes an adverb if translated to English.
- For some multilingual NLP tasks, such as unsupervised dependency parsing (or parser transfer), it might be more important to preprocess all languages under study as similarly as possible (including POS tagging), rather than to maximize accuracy with respect to highly different gold-standard data in individual languages.

We propose "delexicalized tagging", a new method for under-resourced languages. In analogy to delexicalized parsing (Zeman and Resnik, 2008), we transfer a tagging model from a resource-rich language (or a set of languages); the model is independent of individual word forms. In delexicalized parsing, word form sequences are substituted by sequences of POS tags, which—of course—is not extendable to tagging. Instead, we substitute word forms by vectors of numerical features that can be computed using only unannotated monolingual texts. The background intuition is that the individual POS categories will tend to manifest similar statistical properties across languages (e.g., prepositions tend to be short, relatively frequent, showing different patterns of conditional entropy to the left versus to the right, as well as certain asymmetry of occurrences along sentence length). Thus, unlike most POS tagging methods for resource-poor languages, we do not transfer the tagging knowledge using dictionaries or parallel data, but exclusively via the R^n space.[1]

In addition, we present a new publicly available resource containing POS-labeled texts for 107 languages, automatically tagged by the presented approach.

2. Related Work

There is a body of literature about POS tagging of under-resourced languages. Most approaches rely on the existence of some form of parallel (or comparable) data. We will discuss only those approaches that attempt at using the same tagset across languages, and not those aiming at unsupervised induction, such as the well-known Brown clusters induced in a fully unsupervised fashion (Brown et al., 1992). An overview of such truly unsupervised approaches can be found in (Christodouloupoulos et al., 2010).[2]

[1] However, we do not say that our method is completely language-independent. For instance, we rely on the existence of a meaningful tokenization in the target language.

[2] There is a certain terminological confusion in this area: sometimes the word "unsupervised" is used also for situations in which there are no hand-tagged data available for the target language, but some manual annotation of the source language exists and is projected across parallel data like in (Das and Petrov, 2011). We prefer to avoid the term "unsupervised" when manual annotation is used in any language.

(Yarowsky and Ngai, 2001) project POS tags from English to French and Chinese via both automatic and gold alignment, and report substantial growth of accuracy after using de-noising postprocessing. (Fossum and Abney, 2005) extend this approach by projecting multiple source languages onto a target language.

(Das and Petrov, 2011) use graph-based label propagation for cross-lingual knowledge transfer, and estimate emission distributions in the target language using a log-linear model. (Duong et al., 2013) choose only automatically recognized "good" sentences from the parallel data, and further apply self-training.

(Agić et al., 2015) learn taggers for 100 languages using aligned Bible verses from The Bible Corpus (Christodouloupoulos et al., 2010).

Besides approaches based on parallel data, there are also experiments showing that reasonable POS tagging accuracy (close to 90 %) can be reached using quick and efficient prototyping techniques, such as (Cucerzan and Yarowsky, 2002). However, such approaches rely on at least partial understanding of the target language grammar, and on the availability of a dictionary, hence they do not scale well when it comes to tens or hundreds of languages (Cucerzan and Yarowsky experiment with two languages only).

3. Delexicalized Tagging

We propose a statistical method to predict the POS tags in a previously unseen language. The method is quite different from those described above. Our system needs just a raw corpus of the target language—something that can be easily obtained for a large number of world's languages.

3.1. Overview

We proceed as follows:

1. we identify the sets of source languages (those for which we have POS labeled data) and target languages (those for which we have sufficiently big monolingual data and which we want to label by our method),

2. for each word type in the source and target languages, we extract a feature vector that describes its statistical properties in the corresponding monolingual corpus,

3. for all source languages, each word feature vector and its POS tag are used as a training instance for a classifier, and the resulting classifier is used to assign POS tags to all words' feature vectors in the target languages,

4. we evaluate our approach on the target languages for which there are labeled data available, and assume that reasonably similar accuracies are reached also for the other target languages.

3.2. Tagset

A prerequisite to our approach is a common tagset for both the source and the target languages. We use the same tagset as (Das and Petrov, 2011), the Google Universal POS tag set (Petrov et al., 2012). With just 12 tags it is fairly coarse-grained, which is advantageous for a resource-poor method such as ours; nevertheless it has proved useful in downstream applications such as parsing. The 12 tags are `NOUN`, `VERB`, `ADJ` (adjective), `ADV` (adverb), `PRON` (pronoun), `DET` (determiner), `NUM` (numeral), `ADP` (adposition), `CONJ` (conjunction), `PRT` (particle), `PUNC` (punctuation) and `X` (unknown).

This tagset was recently extended in the Universal Dependencies project[3] (Nivre et al., 2016): five categories were split to finer subclasses. Using this larger tagset in our experiments is likely to reduce reliability of the results.

3.3. Features

The list below describes the features that we use for the POS prediction. Let us define our notation first. Let C be a corpus and c_i the i-th token in the corpus. $N = |C| =$ the number of tokens in the corpus C. $f(w) = |\{i : c_i = w\}| =$ the absolute word frequency, i.e. number of instances of the word type w in the corpus C. Similarly, $f(x, y)$ is the absolute frequency of the word bigram xy. $Pre(w) = \{x : \exists i\, (c_i = w) \wedge (c_{i-1} = x)\}$ is the set of word types that occur at least once in a position preceding an instance of w. Analogously, $Next(w)$ denotes the set of word types following w in the corpus. $Context(w) = \{x, y : \exists i\, (c_{i-1} = x) \wedge (c_i = w) \wedge (c_{i+1} = y)\}$ denotes the set of contexts surrounding w, and $Subst(w) = \{y : Context(y) \cap Context(w) \neq \emptyset\}$ is the set of words that share a context with w.

1. *word length* – the number of characters in w

2. *log frequency* – logarithm of the relative frequency of w in C
$$\log \frac{f(w)}{N}$$

3. *preceding word entropy*
$$PN = \sum_{y \in Pre(w)} f(y)$$
$$\sum_{y \in Pre(w)} -\frac{f(y)}{PN} \log \frac{f(y)}{PN}$$

4. *following word entropy*
$$NN = \sum_{y \in Next(w)} f(y)$$
$$\sum_{y \in Next(w)} -\frac{f(y)}{NN} \log \frac{f(y)}{NN}$$

5. *substituting word entropy*
$$SN = \sum_{y \in Subst(w)} f(y)$$
$$\sum_{y \in Subst(w)} -\frac{f(y)}{SN} \log \frac{f(y)}{SN}$$

[3] `http://universaldependencies.org/`

6. *is number* – binary value $is\_number(w)$,

7. *is punctuation* – binary value $is\_punctuation(w)$,

8. *relative frequency after number*

$$\log \frac{|i : c_i = w \wedge is\_number(c_{i-1})|}{f(w)}$$

9. *relative frequency after punctuation*

$$\log \frac{|i : c_i = w \wedge is\_punctuation(c_{i-1})}{f(w)}$$

10. *weighted sum of pointwise mutual information (PMI) of w with the preceding word* – collect all words y in C that precede w, then calculate their PMI values with w and make summation of PMIs weighted by the joint probability of the pair

$$\frac{\sum_{y \in Pre(w)} f(w,y) \times \log \frac{N \times f(w,y)}{f(w) \times f(y)}}{N}$$

11. *weighted sum of PMI of w with the following word* – fully analogous to the previous feature,

12. *entropy of suffixes following the root of w* – First we collect counts of suffixes $count(suffix)$ in C whose length range from 1 to 4 and counts of respective roots (words without suffixes) $count(root)$ in C. For each word, we find the border between root and suffix by maximization of the product $f(root) \times f(suffix)$. Then, we compute conditional entropy over all suffixes given the root.[4]

13. *how many different words appear before w*: $|Pre(w)|$

14. *how many different words appear after w*: $|Next(w)|$

15. *how many different words in C share the same context as w*: $|Subst(w)|$

16. *pointwise mutual information between w and the most frequent preceding word*

$$MaxP = \arg\max_{y \in Pre(w)} f(y)$$

$$\log \frac{N \times f(w, MaxP)}{f(w) \times f(MaxP)}$$

17. *pointwise mutual information between w and the most frequent following word* – fully analogous to the previous feature.

3.4. Data Resources

In our approach, we need two types of data resources:

- raw monolingual texts for both source and target languages; this data is used for extracting feature vectors for words in individual languages; we use W2C, a web-based corpus of 120 languages (Majliš and Žabokrtský, 2012),

- POS-tagged data for source languages; this data is used for training POS classifiers; we use HamleDT 2.0 (Zeman et al., 2014), a collection of treebanks for 30 languages.

3.5. Training POS Classifiers

We took the first 50,000 tokens from the HamleDT 2.0 training sections of 13 languages (ISO 639-1 codes): bg, ca, cs, de, el, en, hi, hu, it, pt, ru, sv, tr. Each token was considered one training instance (i.e., n occurrences of a word w results in n identical instances). Their word feature vectors were computed using at most the first 20 million tokens from the WEB part of the W2C corpus.

We experimented with several types of classifiers:

- **Baseline** We assign `PUNC` to all tokens consisting of non-alphanumerical characters, `NUM` to all tokens containing a digit, and `NOUN` to the remaining tokens.

- **K-nearest-neighbors** (KNN) (Cover and Hart, 1967), with $k = 100$.

- **Support vector machines** (SVM) with radial kernel (Boser et al., 1992).

- **Bagging** (Breiman, 1996) applied both on KNN and SVM. We randomly sampled the training instances with replacement and randomly extracted half of the whole feature space with replacement.

- **Random Forest** (Ho, 1995).

- **Gradient Tree Boosting** (Friedman, 2002).

We trained classifiers for each source language separately and then for concatenated data of the following 7 languages: bg, ca, de, el, hi, hu, tr (in our results, we refer to these combined data as “c7”).

3.6. Evaluation

The first 1000 tokens from the HamleDT 2.0 test sections of the following languages were used for evaluation: bg, bn, ca, cs, da, de, el, es, en, et, eu, fa, fi, hi, hu, it, la, nl, pt, ro, ru, sk, sl, sv, te, tr. Again, feature vectors for individual words are based on the the WEB component of W2C. Naturally there will be words in the test data that have not been observed in W2C. Since we cannot compute the features of these out-of-vocabulary words, we predict their tag as `NUM` if they contain a digit, and as `NOUN` otherwise.

Each target language was evaluated separately for each source language, and then for the above mentioned mixture of 7 languages (c7); The results using the Bagging SVM classifier are summarized in Table 1.

[4] The underlying intution is that some POSs tend to participate in derivation and inflection more intensively than others. Obviously, the root/suffix segmentation is approximated only very roughly here.

target	source														
	bg	ca	cs	de	el	en	hi	hu	it	pt	ru	sv	tr	avg	c7
bg	**86.6**	43.2	**59.0**	53.9	54.9	53.9	53.8	46.2	52.2	58.3	43.0	58.9	40.7	51.5	**75.2**
bn	27.6	34.0	38.7	41.1	26.7	41.5	52.2	36.2	32.3	39.5	23.8	35.7	51.7	37.0	**60.8**
ca	46.9	**84.6**	52.5	47.1	50.8	43.6	45.6	51.2	65.9	**70.0**	37.3	44.3	38.3	49.5	**74.6**
cs	**68.5**	45.4	**84.3**	63.3	56.2	63.3	50.5	58.4	53.2	47.7	54.4	63.7	50.7	56.3	65.6
da	61.7	47.7	52.1	55.1	42.1	66.4	40.1	40.6	50.5	53.0	32.8	**75.0**	41.0	50.6	57.3
de	55.6	49.5	61.9	**91.0**	53.7	**69.9**	46.6	57.7	56.5	59.2	47.4	66.1	53.5	56.5	**83.5**
el	50.5	58.9	49.7	47.9	**87.0**	40.1	38.5	55.2	**65.0**	57.2	42.7	48.3	38.0	49.3	**78.5**
en	54.5	46.8	57.3	60.8	51.5	**86.0**	50.9	46.1	52.2	49.5	41.0	**66.1**	56.1	52.7	62.6
es	58.8	74.6	49.6	47.7	61.6	54.5	51.3	52.1	75.4	**79.8**	37.3	50.8	38.7	56.3	67.5
et	53.7	39.0	59.3	57.1	45.7	41.9	38.9	54.9	51.0	44.8	39.2	58.3	54.2	49.1	**64.1**
eu	35.7	41.3	47.0	57.2	34.6	48.4	46.7	46.8	39.5	43.6	22.1	47.1	54.5	43.4	**62.0**
fa	37.6	41.4	46.9	49.2	33.9	49.7	65.4	25.3	42.5	42.7	37.2	39.5	54.8	43.5	**65.9**
fi	43.9	27.8	**51.4**	46.8	41.3	37.4	41.3	45.5	38.5	30.6	37.1	45.3	50.3	41.3	**51.4**
hi	48.6	**63.1**	40.3	40.2	31.2	55.0	**90.6**	31.7	47.8	40.2	46.8	38.8	41.8	43.8	**86.5**
hu	44.0	54.4	**57.6**	53.8	54.5	38.7	37.2	**81.2**	52.1	50.8	35.2	49.7	50.6	48.2	**73.5**
it	58.2	67.3	59.0	58.2	62.0	61.4	49.1	51.1	**88.5**	**70.8**	47.1	54.7	44.1	56.9	70.2
la	30.4	28.0	49.7	43.5	32.4	36.7	39.3	39.6	31.7	26.1	41.9	37.5	49.7	37.4	**51.1**
nl	53.0	54.0	55.0	**66.1**	56.8	56.0	40.9	62.0	62.2	59.1	40.4	58.3	41.4	54.2	60.0
pt	61.9	55.1	50.2	51.8	49.7	48.1	47.7	54.9	**74.4**	**84.9**	43.0	48.6	41.8	52.3	65.1
ro	50.9	42.3	46.7	50.0	43.1	52.4	57.1	42.9	**62.9**	59.3	54.8	39.6	41.1	49.5	57.2
ru	45.2	22.9	**51.5**	40.8	33.7	36.4	44.6	38.1	37.3	30.1	**70.8**	40.0	37.7	38.2	43.4
sk	60.6	38.2	**70.7**	54.6	46.6	44.4	41.7	44.8	44.2	46.8	45.8	51.8	41.4	48.6	56.0
sl	59.1	41.0	58.9	55.1	48.4	47.9	35.9	45.8	53.3	49.3	30.1	**61.3**	44.6	48.5	59.4
sv	63.3	46.8	56.5	62.1	45.0	**64.5**	39.5	45.0	50.4	50.8	43.3	**80.5**	41.9	50.8	63.0
te	28.0	26.0	39.5	59.3	26.8	41.0	49.9	41.2	32.0	40.7	33.7	37.0	**62.3**	39.8	57.0
tr	28.2	26.5	41.8	**48.8**	24.2	37.4	39.0	44.0	26.9	33.3	26.7	33.4	**77.6**	34.2	**70.9**

Table 1: POS tagging accuracy using bagging based on SVM. **Highlighted** results indicate that the same language was used for training and testing. **Bold** indicates the best result where the target language was not used in training. The *avg* column shows the average accuracy for given language (not counting the highlighted results). The *c7* training data stands for the concatenation of 7 source languages: bg, ca, de, el, hi, hu, and tr.

Table 2 compares the scores of different classifiers. All the classifiers were trained on $c7$; the languages included in $c7$ were excluded from the testing set. The standard SVM classifier performs better than KNN, the average tagging accuracy on $c7$ is 4.7% higher and it is better on 15 out of 19 languages. Bagging improves the average accuracy of KNN by 3%. The SVM's average accuracy slightly decreases when bagging is used, however, 9 out of 19 languages are tagged better. We observed improvement for both classifiers also over models trained on individual languages. The *Gradient tree boosting* classifier is by 1.4% worse than SVM.

(Ho, 1998) suggests to expand feature vectors by using certain functions of the original features (e.g., pairwise summation, pairwise differences, pairwise products and boolean combination for binary and categorical features). For the *Random forest classifier*, we used the same techniques as bagging (sampled both instances and features with replacement) and we also expanded the feature space from 17 to 20 features using feature combination methods.[5] Even though the combined features do not contribute new information, being able to weigh their concurrent appearance actually increases accuracy.

4. Error Analysis

Table 3 shows the confusion matrix of tag prediction. It is no surprise that punctuation (`PUNC`) is the easiest category to predict. At the other end of the scale, the `X` category will intuitively contain words of mixed nature, which is impossible to predict.

Certain idiosyncracies of tokenization schemes negatively affect the results. The underscore (“_”) token is ex-

[5] Here we used word frequency + number of distinct preceding words, word frequency + number of distinct following words, word frequency + number of distinct words sharing the same context

lang	baseline	KNN	Bagging KNN	SVM	Bagging SVM	Gradient TreeBoost	Random Forest
bn	48.0	54.6	55.0	52.8	**60.8**	60.3	56.4
cs	54.9	63.6	65.2	**68.3**	65.6	65.9	67.9
da	37.0	53.7	57.9	**62.9**	57.3	62.5	60.2
en	45.7	58.7	60.0	**67.3**	62.6	62.0	59.7
es	36.9	63.3	69.1	**68.0**	67.5	65.1	66.1
et	53.3	57.3	58.8	61.8	**64.1**	57.5	57.5
eu	44.7	56.4	50.3	55.2	**62.0**	51.3	53.8
fa	50.6	59.8	59.8	61.9	**65.9**	58.8	48.0
fi	**51.6**	47.4	49.6	50.7	51.4	46.2	46.2
it	43.1	65.7	70.9	**74.1**	70.2	71.2	70.5
la	47.2	43.7	47.8	50.0	**51.1**	46.5	50.5
nl	33.2	68.8	66.6	**74.2**	60.0	70.9	67.7
pt	38.8	63.9	**66.6**	57.9	65.1	66.0	64.4
ro	40.2	48.6	57.5	**61.9**	57.2	48.1	52.9
ru	40.0	39.2	43.3	**45.0**	43.4	44.0	43.3
sk	45.6	50.4	51.8	**58.5**	56.0	57.1	56.7
sl	40.8	53.7	58.0	58.0	**59.4**	67.6	64.6
sv	38.0	54.0	60.8	**65.0**	63.0	60.8	63.4
te	45.6	55.6	57.0	54.4	57.0	**59.5**	54.6
avg	44.0	55.7	58.7	**60.4**	60.0	59.0	58.1

Table 2: Results of different classifiers and their average. All classifiers in this table were trained on c7 (combination of bg, ca, de, el, hi, hu, and tr), and they were evaluated on languages outside of c7.

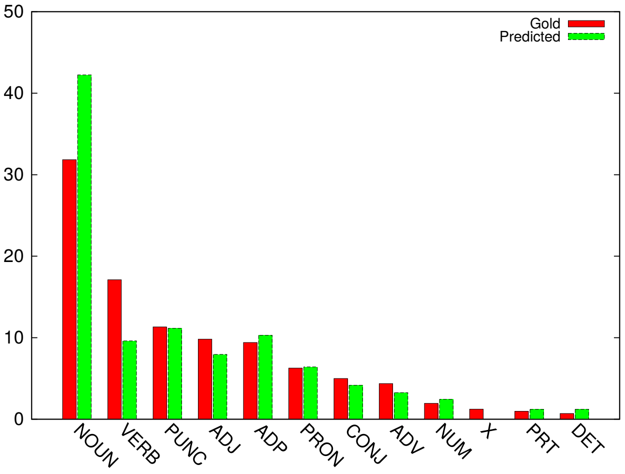

Figure 1: Distribution of manually assigned POS tags and predicted POS tags. The numbers are computed over all the 19 testing corpora (i.e., excluding $c7$).

	NO	VB	AJ	AV	PR	DT	NU	AP	CJ	PT	PU	X
NO	**6811**	452	402	96	137	9	29	47	7	10	7	
VB	1440	**1862**	229	107	115		8	195	42	138		
AJ	1240	123	**1073**	37	222	1	46	22	2	6		
AV	255	85	84	**344**	156	6	5	59	37	7		
PR	138	83	150	109	**684**	40	82	120	92	16		
DT	1	1	87	3	6	**191**	14	8				
NU	64	17	27	1	34	3	**364**	6	1			
AP	150	40	142	32	21	105	2	**1927**	40	6	4	
CJ	33	10	6	40	106	29		162	**816**	23	3	
PT	19	7	1	18	45	27		49	15	**119**		
PU	72										**2760**	
X	159	7	5	4	7		27	6			24	**2**

Table 3: Confusion matrix of the best classifier, evaluated on all target languages (sum). Rows correspond to gold-standard tags, columns to predicted tags. NO = NOUN; VB = VERB; AJ = ADJ; AV = ADV; PR = PRON; DT = DET; NU = NUM; AP = ADP; CJ = CONJ; PT = PRT; PU = PUNC.

tensively used in Catalan, Spanish (dropped pronominal subjects) and in Turkish (representing stages of morphological derivation). Hindi has empty NULL nodes (often but not always representing elided verbs). Several languages contain multi-word expressions collapsed into one token (e.g. [es] *Tribunal_Supremo_de_Justicia*); since these are not naturally occurring strings, they are out of our vocabulary (they have no footprint in the W2C corpus).

We could remove the "NULL" and "_" nodes from both training and testing data to get results that are closer to real-world application. Note however that we cannot automatically split the multi-word expressions because we do not have gold tags for the individual words.

The model is quite successful in predicting prepositions (ADP), conjunctions (CONJ), nouns (NOUN; this is the most frequent part of speech in most languages, hence our recall is significantly higher than precision) and numerals (NUM; numbers expressed by digits, which are as easy as punctuation, help to boost this category).

On the other hand, the model is unsuccessful in predicting adjectives, adverbs, pronouns and particles. For particles (PRT) the explanation could be that they are poorly defined, or their definition significantly differs across languages.

A better definition or even partition of pronouns may

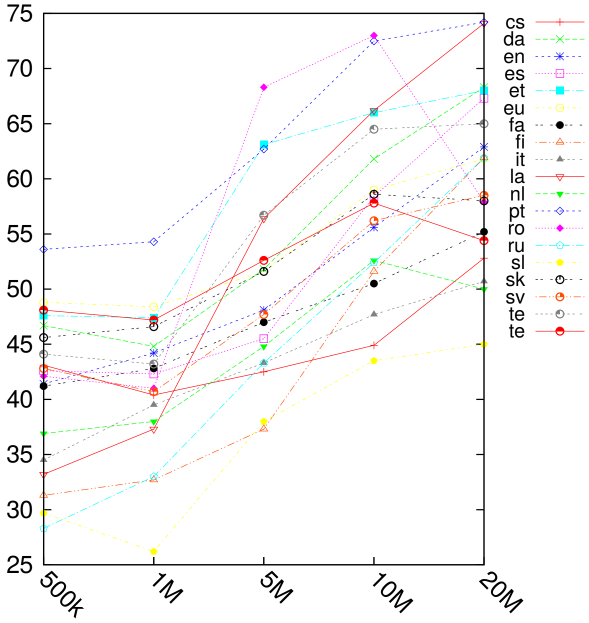

Figure 2: Learning curves for different sizes of texts, on which the features for individual test-set words were computed.

also help: personal pronouns do not occur in the same contexts as possessive or relative pronouns, and languages also disagree in the pronoun-determiner distinction. Furthermore, pro-drop languages use personal pronouns much more sparingly than e.g. English or German. Similarly, many languages lack articles (tagged `DET`). As is apparent from the c7 output, articles such as English *a* and *the* often end up tagged as adjectives. That seems a good back-off decision because articles modify noun phrases similarly to adjectives.

Obviously, we can improve the results if we know something about the target language and if one or more related languages are available in our source data.

Example 1: target Portuguese. When trained on c7, the tagging accuracy is 58%; when trained on Italian, the accuracy jumps to 71%, in spite of the training data being 7 times smaller. One of the c7 languages is Catalan, supposedly close to Portuguese, but the other languages introduce too much noise. Detailed analysis reveals that the Italian model dramatically improves recall of adjectives and prepositions, and precision of numerals. Verbs rise in recall and drop in precision but the F score is still better than with c7. On the other hand, the recall of pronouns is seriously damaged as only 11/72 are correctly identified (while it was 30/72 with c7).

Example 2: target Slovak. When trained on c7, the tagging accuracy is 59%; when trained on Czech, the accuracy jumps to 75%. The only Slavic language in c7 is Bulgarian, and it is an outlier among Slavic languages because it has lost the case system of nouns. Detailed analysis reveals that it is extremely difficult (for both models) to distinguish Slovak adverbs from nouns. On the other hand, prepositions are moderately difficult with the c7 model (P=48%, R=75%) but they are practically solved with the Czech model (P=R=99%). The c7 model mistook many pronouns and other short closed-class words for prepositions. Pronouns, that are in general quite difficult to predict, have poor results with the c7 model (F=20%) but they come quite well with the Czech model (F=79%).

Example 3: target Basque. Basque is an isolated language, without known genetic relationship to any other language. It is an agglutinating language with a comparatively rich case system, so one might be tempted to choose Hungarian as the source language. But the accuracy (see Table 1) would be less than 47%. The best single-source result is yielded by German (57%), which superficially resembles agglutinating languages with its long compound words. Nevertheless, the mixed model proves to be the best source for isolated languages like Basque: the best accuracy, 62%, is achieved with the c7 model.

5. Deltacorpus

The configuration that performs best, which is the SVM classifier trained on the mixture of 7 source languages, was used to tag texts in 107 languages selected from the W2C corpus, 1 million tokens per language. This new resource is called *Deltacorpus* (a corpus tagged by a DELexicalized TAgger) and it is available on-line[6] under the CC BY-SA license. Table 4 gives a summary of the languages. We have excluded languages whose WEB corpus in W2C is too noisy (especially due to wrong language identification), as well as a few Asian languages with non-trivial word segmentation (e.g., Chinese, Japanese and Thai).

6. Conclusions and Future Work

This paper presents a new method for cross-language transfer of POS-tagging models. To the best of our knowledge, this is the first attempt at transferring POS taggers without any bilingual (parallel or comparable) data. We experimented with various language-independent features and several classifiers; the SVM with 17 features, trained on a mix of 7 languages, outperformed other models on our evaluation data.

In most cases, the tagging accuracy improved over the baseline. We thus conclude that human-defined word categories naturally incline towards properties which may give them away even in a totally unknown language. The performance is well below results achieved by contemporary methods based on parallel data, however, it is completely independent of the existence of any parallel or comparable corpora or dictionaries.

We released Deltacorpus, a collection of texts in 107 languages tagged by the best classifier, assuming that the tagging accuracy will be comparable to what we observed on our evaluation data. For the sake of completeness, we have also included languages for which better resources exist. However, there are dozens of languages that are not even represented in the Bible corpus. We believe that for these languages Deltacorpus can provide a temporary solution, until more resources are available.

In the future we plan to implement several natural extensions of our approach. For instance, we currently disregard that a word form can have multiple readings, we even

[6] `http://hdl.handle.net/11234/1-1662`

Code	Name	Family	BibC
bel	Belarusian	IE / Slavic	
bos	Bosnian	IE / Slavic	
bg, bul	Bulgarian	IE / Slavic	yes
cs, ces	Czech	IE / Slavic	yes
hbs	Serbo-Croatian	IE / Slavic	
hrv	Croatian	IE / Slavic	yes
hsb	Upper Sorbian	IE / Slavic	
mkd	Macedonian	IE / Slavic	
pl, pol	Polish	IE / Slavic	yes
ru, rus	Russian	IE / Slavic	yes
sk, slk	Slovak	IE / Slavic	yes
sl, slv	Slovenian	IE / Slavic	yes
srp	Serbian	IE / Slavic	yes
ukr	Ukrainian	IE / Slavic	yes
lav	Latvian	IE / Baltic	yes
lit	Lithuanian	IE / Baltic	yes
afr	Afrikaans	IE / Germanic	yes
da, dan	Danish	IE / Germanic	yes
de, deu	German	IE / Germanic	yes
en, eng	English	IE / Germanic	yes
fao	Faroese	IE / Germanic	
fry	Frisian	IE / Germanic	
gsw	Alemannic	IE / Germanic	
isl	Icelandic	IE / Germanic	yes
lim	Limburgish	IE / Germanic	
ltz	Luxembourgish	IE / Germanic	
nds	Low Saxon	IE / Germanic	
nl, nld	Dutch	IE / Germanic	
nno	Nynorsk	IE / Germanic	
nor	Norwegian	IE / Germanic	yes
sco	Scots	IE / Germanic	
sv, swe	Swedish	IE / Germanic	yes
yid	Yiddish	IE / Germanic	
arg	Aragonese	IE / Romance / Italic	
ast	Asturian	IE / Romance / Italic	
ca, cat	Catalan	IE / Romance / Italic	
fra	French	IE / Romance / Italic	yes
glg	Galician	IE / Romance / Italic	
hat	Haitian Creole	IE / Romance / Italic	yes
it, ita	Italian	IE / Romance / Italic	yes
la, lat	Latin	IE / Romance / Italic	yes
lmo	Lombard	IE / Romance / Italic	
nap	Neapolitan	IE / Romance / Italic	
pms	Piedmontese	IE / Romance / Italic	
pt, por	Portuguese	IE / Romance / Italic	yes
ro, ron	Romanian	IE / Romance / Italic	yes
es, spa	Spanish	IE / Romance / Italic	yes
vec	Venetian	IE / Romance / Italic	
wln	Walloon	IE / Romance / Italic	
bre	Breton	IE / Celtic	
cym	Welsh	IE / Celtic	
gla	Scottish Gaelic	IE / Celtic	yes
gle	Irish	IE / Celtic	
el, ell	Greek	IE /	yes
hye	Armenian	IE /	yes
sqi	Albanian	IE /	yes
diq	Dimli	IE / Iranian	
fa, fas	Persian	IE / Iranian	yes
glk	Gilaki	IE / Iranian	
kur	Kurdish	IE / Iranian	
tgk	Tajik	IE / Iranian	
bn, ben	Bengali	IE / Indo-Aryan	
bpy	Bishnupriya	IE / Indo-Aryan	
guj	Gujarati	IE / Indo-Aryan	
hif	Fiji Hindi	IE / Indo-Aryan	
hi, hin	Hindi	IE / Indo-Aryan	yes
mar	Marathi	IE / Indo-Aryan	yes
nep	Nepali	IE / Indo-Aryan	yes
urd	Urdu	IE / Indo-Aryan	
amh	Amharic	AA / Semitic	yes
ar, ara	Arabic	AA / Semitic	yes
arz	Egyptian Arabic	AA / Semitic	
heb	Hebrew	AA / Semitic	yes
et, est	Estonian	Uralic / FinUgric	yes
fi, fin	Finnish	Uralic / FinUgric	yes
hu, hun	Hungarian	Uralic / FinUgric	yes
eu, eus	Basque		yes
kat	Georgian	Caucasian	
chv	Chuvash	Turkic / Oghur	
aze	Azerbaijani	Turkic / Oghuz	
tr, tur	Turkish	Turkic / Oghuz	
uzb	Uzbek	Turkic / Karluk	
kaz	Kazakh	Turkic / Kipchak	
tat	Tatar	Turkic / Kipchak	
sah	Yakut	Turkic / Siberian	
kor	Korean	Altaic	yes
mon	Mongol	Altaic	
te, tel	Telugu	Dravidian	yes
kan	Kannada	Dravidian	yes
mal	Malayalam	Dravidian	yes
ta, tam	Tamil	Dravidian	
new	Newar	Sino-Tibetan	
vie	Vietnamese	Austroasiatic	yes
ind	Indonesian	Austronesian	yes
jav	Javanese	Austronesian	
mlg	Malagasy	Austronesian	yes
mri	Maori	Austronesian	yes
msa	Malay	Austronesian	
pam	Pampangan	Austronesian	
sun	Sundanese	Austronesian	
tgl	Tagalog	Austronesian	yes
war	Waray	Austronesian	
swa	Swahili	NC / Bantu	yes
epo	Esperanto	constructed	yes
ido	Ido	constructed	
ina	Interlingua	constructed	
vol	Volapük	constructed	

Table 4: The 107 languages in Deltacorpus. Languages from W2C (target languages) are identified by their ISO 639-3 code. Two-letter codes are used to identify languages in HamleDT (source languages). Language family abbreviations: IE = Indo-European, AA = Afro-Asiatic, NC = Niger-Congo. The BibC column tells whether the language is present in the Bible Corpus.

disregard local context in sentences to be tagged, and we do not do any weighting of languages according to their similarity or genealogical relatedness. Above all, we would like to explore possible combinations of our approach with the state-of-the-art techniques based on parallel corpora, as we find them complementary. We also plan on releasing a new version of Deltacorpus where the classifiers will be trained on Universal Dependencies treebanks; as a successor of HamleDT, UD should be better harmonized and more reliable.

Acknowledgments

The work was supported by the grants 14-06548P and 15-10472S of the Czech Science Foundation, by the EU project H2020-ICT-2014-1-644402 (HimL), and by the LINDAT/CLARIN project No. LM2015071 of the Ministry of Education, Youth, and Sports of the Czech Republic.

7. Bibliographical References

Agić, Ž., Hovy, D., and Søgaard, A. (2015). If all you have is a bit of the bible: Learning pos taggers for truly low-resource languages. In *The 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2015).*

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, New York, NY, USA. ACM.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140, August.

Brown, P. F., deSouza, P. V., Mercer, R. L., Della Pietra, V. J., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, December.

Christodouloupoulos, C., Goldwater, S., and Steedman, M. (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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13:21– 27.

Cucerzan, S. and Yarowsky, D. (2002). Bootstrapping a multilingual part-of-speech tagger in one person-day. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Das, D. and Petrov, S. (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*, pages 600–609, Portland, Oregon, USA, June. Association for Computational Linguistics.

Duong, T., Bird, S., Cook, P., and Pecina, P. (2013). Simpler unsupervised POS tagging with bilingual projections. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, number Volume 2: Short Papers, pages 634–639, Sofija, Bulgaria. Bălgarska akademija na naukite, Association for Computational Linguistics.

Fossum, V. and Abney, S. P. (2005). Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In Robert Dale, et al., editors, *Natural Language Processing - IJCNLP 2005, Second International Joint Conference, Jeju Island, Korea, October 11-13, 2005, Proceedings*, volume 3651 of *Lecture Notes in Computer Science*, pages 862–873. Springer.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, February.

Ho, T. K. (1995). Random decision forests. In *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1*, ICDAR '95, pages 278–, Washington, DC, USA. IEEE Computer Society.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

Majliš, M. and Žabokrtský, Z. (2012). Language richness of the web. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2927–2934, İstanbul, Turkey. European Language Resources Association.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia. European Language Resources Association.

Petrov, S., Das, D., and McDonald, R. (2012). A universal part-of-speech tagset. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, pages 2089–2096, İstanbul, Turkey. European Language Resources Association.

Yarowsky, D. and Ngai, G. (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, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeman, D. and Resnik, P. (2008). Cross-language parser adaptation between related languages. In *Workshop on NLP for Less-Privileged Languages, IJCNLP*, Hyderabad, India.

Zeman, D., Dušek, O., Mareček, D., Popel, M., Ramasamy, L., Štěpánek, J., Žabokrtský, Z., and Hajič, J. (2014). HamleDT: Harmonized multi-language dependency treebank. *Language Resources and Evaluation*, 48(4):601–637.