Planting Trees in the Desert: Delexicalized Tagging and Parsing Combined

Daniel Zeman,⋆ David Mareček,⋆ Zhiwei Yu,† and Zdeněk Žabokrtský⋆
⋆ Charles University, Prague, Czechia
† Shanghai Jiaotong University, Shanghai, China
{zeman|marecek|zabokrtsky}@ufal.mff.cuni.cz
jordanyzw@sjtu.edu.cn

Abstract

Various unsupervised and semi-supervised methods have been proposed to tag and parse an unseen language. We explore delexicalized parsing, proposed by (Zeman and Resnik, 2008), and delexicalized tagging, proposed by (Yu et al., 2016). For both approaches we provide a detailed evaluation on Universal Dependencies data (Nivre et al., 2016), a de-facto standard for multi-lingual morpho-syntactic processing (while the previous work used other datasets). Our results confirm that in separation, each of the two delexicalized techniques has some limited potential when no annotation of the target language is available. However, if used in combination, their errors multiply beyond acceptable limits. We demonstrate that even the tiniest bit of expert annotation in the target language may contain significant potential and should be used if available.

1 Introduction

Dependency parsing is an important step in language analysis, useful for downstream applications such as machine translation or question answering. Unfortunately, it is not an easy task. Successful parsers rely on dependency treebanks annotated by language experts. While at least small treebanks are becoming available for an increasing number of languages, the world's languages will not be covered any soon. The number of languages for which at least a small treebank is available lies probably somewhere between 50 and 100 (we are aware of treebanks for 56 languages). At the same time, the number of world's languages is usually estimated between 4,000 and 7,000; and 398 languages are reported to have more than 1 million speakers (Lewis et al., 2016). In order to parse the treebankless languages, several techniques have been developed.

(Hwa et al., 2004) projected dependency trees across bilingual word alignments in a parallel corpus. They used a few target-language rules to improve the target trees.

(Zeman and Resnik, 2008) proposed *delexicalized parsing*, a method that trains a parsing model on part-of-speech tags only, ignoring lexical information. The trained model is then used to parse data in a related language for which POS tags are available. It is assumed that POS-tagged data are cheaper and easier to obtain for new languages than treebanks are. Such claim is probably justified, yet it does not provide any immediate solution in the case that no annotated resources are available for the target language.

(McDonald et al., 2011) evaluated their multi-source delexicalized transfer using POS tags predicted by the projected part-of-speech tagger of (Das and Petrov, 2011). This tagger relies only on labeled training data for English, and uses a parallel corpus (Europarl) to project the tags across word alignment. Both (Zeman and Resnik, 2008) and (McDonald et al., 2011) notice that varying treebank annotation styles are a major obstacle to meaningful evaluation of any cross-linguistic transfer.

Projection across bitexts is the central approach in many published experiments with POS tagging of low-resource languages.

(Yarowsky and Ngai, 2001) project POS tags

from English to French and Chinese via both automatic and gold alignment, and report substantial improvement of accuracy after using de-noising post-processing. (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 loglinear 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).

In contrast, (Yu et al., 2016) train a tagging model on language-independent meta-features and transfer it directly to a target language in a fashion similar to the delexicalized parsing; they call their approach *delexicalized tagging*. They use neither parallel corpora nor any target-language dictionary, rules or other expert knowledge. They compute meta-features on large raw corpora, and they make tagged texts of 107 languages available for download.[1]

2 Delexicalized Tagging

(Yu et al., 2016) describe 17 features they extract for each word type in each source and target language. The features describe statistical properties of the word type in a large raw corpus. They are not directly tied to the lexicon of any particular language.

Languages for which POS-tagged data is available can be used as source languages. A classifier is trained to learn the correspondence between feature vectors and POS tags. The classifier is then directly applied to feature vectors of the target language, and assigns a POS tag to each target word type.

(Yu et al., 2016) experiment with various classifiers and report that *support vector machines* (SVM) with radial kernel (Boser et al., 1992) gave the best results on their data; therefore we use SVM in our experiments, too.

A prerequisite to delexicalized tagging is a common tagset for both the source and the target languages. (Yu et al., 2016) use the Google Universal POS (set of 12 tags) (Petrov et al., 2012). We use an extended version of this tagset, used in the Universal Dependencies project[2] (Nivre et al., 2016). With 17 tags it is still reasonably coarse-grained, which is advantageous for such a resource-poor method.

The 17 tags are `NOUN`, `PROPN` (proper noun), `VERB`, `AUX` (auxiliary verb), `ADJ` (adjective), `ADV` (adverb), `PRON` (pronoun), `DET` (determiner), `NUM` (numeral), `ADP` (adposition i.e. pre- or postposition), `CONJ` (coordinating conjunction), `SCONJ` (subordinating conjunction), `PART` (particle), `INTJ` (interjection), `SYM`, `PUNCT` and `X` (unknown).

2.1 Features

We use the same 17 features[3] as (Yu et al., 2016), which we describe below. 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

[1] Note that a pre-requisity of delexicalized tagging is that word boundaries in the target text are easily detectable. Hence the method is not suitable for languages that do not use inter-word spacing, such as Chinese, Japanese or Thai.

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

[3] Software at `github.com/ufal/deltacorpus`.

2. *log frequency* – logarithm of the relative frequency of w in C

$$\log \frac{f(w)}{N}$$

3. *is number* – binary value based on the Unicode character property *digit*

4. *is punctuation* – binary value based on the Unicode character property *punctuation*

5. *relative frequency after number*

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

6. *relative frequency after punctuation*

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

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

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

9. *how many different words in* C *share a context with* w: $|Subst(w)|$

10. *preceding word entropy*

$$PN = \sum_{y \in Pre(w)} f(y)$$

$$\sum_{y \in Pre(w)} -\frac{f(y)}{PN} \log \frac{f(y)}{PN}$$

11. *following word entropy*

$$NN = \sum_{y \in Next(w)} f(y)$$

$$\sum_{y \in Next(w)} -\frac{f(y)}{NN} \log \frac{f(y)}{NN}$$

12. *substituting word entropy*

$$SN = \sum_{y \in Subst(w)} f(y)$$

$$\sum_{y \in Subst(w)} -\frac{f(y)}{SN} \log \frac{f(y)}{SN}$$

13. *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 sum PMIs weighted by the joint probability of the pair

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

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

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

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

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

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

17. *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 corresponding 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]

3 Delexicalized Parsing

The idea of delexicalized parsing is that a given sequence of parts of speech has often the same preferred dependency structure regardless of language. To illustrate this, consider the bilingual example in Figure 1. The mapping between the words in the two sentences is not 1-1. However, the words they have in common have identical part-of-speech tags and the dependency relations are also shared.

The POS tags are the key here. We can remove the words from the training data and show only the

[4]The underlying intution is that some POSs tend to participate in derivation and inflection more often than others. Obviously, our root/suffix segmentation is only an approximation.

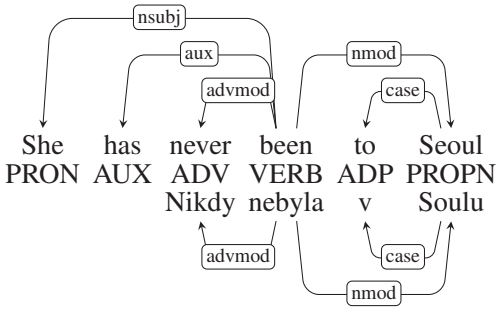

Figure 1: English and Czech sentence with equivalent meaning and shared dependencies.

POS tags to the parser when training the parsing model. Such model will be less accurate because it will lack important lexical information such as verb valency. However, at the same time it will be applicable to multiple languages. Obviously, the more closely related the languages are to the source (training) language, the better.

While a common tagset is a prerequisite to delexicalized tagging, for parsing we assume a common dependency style, i.e. the set of relation types and guidelines for their attachment.[5] Fortunately, Universal Dependencies define a harmonized annotation style for all languages and we do not have to consider annotation differences, unlike (McDonald et al., 2011) and other previous work.

4 Data and Experimental Setup

In order to increase comparability of our results to (Yu et al., 2016), we use the same W2C corpus (Majliš and Žabokrtský, 2012) to extract feature vectors (we use at most the first 20 million tokens from the WEB section in each language).

Unlike (Yu et al., 2016), we take the POS-tagged data for SVM training and evaluation from the Universal Dependencies collection (Nivre et al., 2016). UD is also used to train and evaluate delexicalized parsers. We work with the following 29 UD languages:[6] Basque (eu), Bulgarian (bg), Croatian (hr), Czech (cs), Danish (da), Dutch (nl), English (en), Estonian (et), Finnish (fi), French (fr), German (de), Greek (el), Hebrew (he), Hindi (hi), Hungarian (hu), Indonesian (id), Irish (ga), Italian (it), Latin (la), Norwegian (no), Persian (fa), Polish (pl), Portuguese (pt), Romanian (ro), Slovenian (sl), Spanish (es), Swedish (sv), Tamil (ta). For experiments that do not involve delexicalized tagging we also report results on Ancient Greek (grc), Arabic (ar), Gothic (got) and Old Church Slavonic (cu).[7]

The first 30,000 tokens of the training data of each language were used to train the SVM classifier for delexicalized tagging. Each token was considered one training instance (i.e., n occurrences of a word w result in n identical instances). In addition we trained several mixed models based on multiple source languages ($N \times 30,000$ tokens). (Yu et al., 2016) observe that it is really significant how much similar the source and the target languages are. Hence we trained specialized models for several groups of Indo-European languages (Germanic – ger, Romance – rom, and Slavic – sla), one model of agglutinating languages (agl: Hungarian, Finnish, Estonian, Basque), one general model for Indo-European languages (ine) and one model based on all languages. We always excluded the target language from the source mix. While the first three groups are motivated by genetic relatedness, the *agl* group is based on surface properties because we have few related languages in the collection.

In the case of delexicalized parsing, we compare several scenarios, depending on what is the source of the part-of-speech tags used in parsing models: gold-standard, predicted by a supervised tagger and predicted by a delexicalized tagger. To speed up experimentation, the parser is trained only on the first 5,000 sentences of the training section of the given language. In the case of multi-source transfer, all the source languages are merged first (interlaced, so that all languages can participate), then the first 5,000 sentences are taken. We use the Malt Parser (Nivre and Hall, 2005) with the `stacklazy` algorithm; it is reasonably fast to train and it allows for non-projective dependencies.

[5]The common style is only required for evaluation of the results using some labeled data. When the technique is applied to a truly unknown language, the target annotation style will be naturally inherited from the source language.

[6]We use the UD release 1.2 and exclude languages that are not represented in W2C and also Arabic (because of vowel diacritics and tokenization in UD not matching W2C) and Japanese (because of its non-trivial word segmentation).

[7]There are languages with more than one treebank and we use numeric indices to distinguish the extra treebanks in our results: fi_1 for the FinnTreeBank, la_1 and grc_1 for the PROIEL treebanks, and la_2 for the Index Thomisticus Treebank.

	source								
target	base	self	all	ine	ger	rom	sla	agl	c7
bg	37	87	57	58	61	56	**67**	45	50
cs	42	82	60	59	52	46	**63**	45	54
da	30	83	63	67	**70**	58	48	42	55
de	30	83	58	60	**63**	58	46	39	51
el	36	88	**62**	**62**	51	54	51	40	55
en	30	81	56	58	**62**	54	49	48	52
es	31	89	69	72	72	**79**	56	40	61
et	49	73	**51**	48	47	37	47	**51**	45
eu	42	78	**52**	45	41	40	42	46	44
fa	**48**	89	**47**	39	34	27	**47**	42	42
fi	41	74	**49**	**49**	46	42	**49**	**49**	43
fi_1	35	73	46	47	40	36	42	**48**	41
fr	33	89	73	73	63	**78**	52	41	65
ga	38	84	**58**	56	57	51	49	45	57
he	38	81	**42**	37	34	29	39	34	38
hi	30	85	**51**	45	50	43	46	35	48
hr	41	84	56	60	51	41	**65**	49	52
hu	37	79	51	49	**55**	51	43	37	46
id	42	82	**50**	47	42	41	47	42	44
it	34	86	58	51	56	**74**	53	38	55
la	30	74	**49**	39	34	26	40	43	43
la_1	20	79	**35**	28	23	17	33	22	29
la_2	37	89	50	47	44	39	52	**53**	50
nl	28	83	62	61	**65**	60	44	44	61
no	29	87	66	65	**70**	56	48	44	49
pl	45	80	60	59	48	47	**64**	46	53
pt	36	89	67	67	62	**75**	60	40	51
ro	36	75	**58**	56	47	55	50	39	51
sl	36	78	**62**	60	52	45	**62**	45	53
sv	30	80	65	68	**71**	53	53	46	59
ta	**42**	69	**37**	33	30	25	35	36	32

Table 1: POS tagging accuracy using the SVM classifier, measured on UD 1.2 development data. The "base" column shows baseline results. The "self" column contains results of a classifier trained on the target language. In the remaining columns the target language was always excluded from the source language set. Various combinations of source languages were tested: all, Indo-European, Germanic, Romance, Slavic, a mix of agglutinating languages, and the "c7" combination from (Yu et al., 2016) (but UD 1.2 does not contain Catalan and Turkish, so our mix contains a maximum of 5 languages, minus the source: bg, de, el, hi, hu).

5 Evaluation

Table 1 summarizes the results of delexicalized tagging on UD 1.2 development data. The "base" column presents results of a baseline tagger that tags everything except punctuation and numbers as NOUN. The general tendency is that Romance and Germanic languages, with their higher proportion of function words, have lower baseline accuracy than Slavic and Uralic languages. At the same time, languages with low baseline score often (but not always) witness high accuracy of the SVM tagger. For high-baseline languages the classifier brings only moderate improvement, and in two cases (Persian and Tamil) it does not beat the baseline at all.

The "self" column gives results of a classifier trained on the target language (but training data is still different from test data). It can be understood as an estimate of the upper bound of achievable results.

The rest of the table shows classifiers trained on various combinations of source languages. The grouping is done the same way as for tagging, although there are other options, e.g. the KLcpos3 metric proposed by (Rosa and Žabokrtský, 2015). We have confirmed the hypothesis that if there are multiple closely related source languages available, the more distant languages are better left out. All Slavic, Germanic and Romance languages (except Romanian) achieve the best scores with classifiers trained on their respective groups (the target language always excluded from training). This can be explained by different distribution of parts of speech: Slavic languages do not have articles, hence the 'DET' tag is much less frequent than in Germanic and Romance languages. The replacement of case morphology by prepositions is even more pervasive in Romance than in Germanic languages. And so on.

For the other target languages, training on all available source languages seems to be the best recipe in most cases. For example Hindi, an Indo-European language, is not so close to its relatives (at least w.r.t. the features that we measure) that we could base the classifier solely on Indo-European languages from our collection. However, using the labeled Hindi data to tag Urdu, Punjabi or Gujarati is likely to be more successful.

Table 2 summarizes the unlabeled attachment score (UAS) of delexicalized parsing based on POS tags predicted by the supervised tagger from UDPipe (Straka et al., 2016). There are three exceptions: the columns "gold", "lex" and "l20" use lexicalized parsing models. In addition, "gold" uses gold-standard POS tags.

For some languages the difference between us-

	self				source				
target	gold	lex	delex	l20	single best			multi best	
ar	80	80	70	**58**	**he:50**	got:49	pl:48	sla:45	rom:45
bg	84	88	84	51	sl:73	cs:72	hr:70	**sla:76**	ine:71
cs	81	80	73	55	**hr:62**	bg:60	sl:60	sla:61	ine:60
cu	87	83	77	60	**got:73**	grc_1:69	la_1:57	ger:68	ine:67
da	84	78	72	54	no:63	bg:61	sv:59	**ger:66**	ine:66
de	81	75	67	51	sv:54	sl:53	bg:53	**sla:59**	ger:57
el	80	80	73	60	hr:59	sl:59	ro:51	**sla:63**	ine:62
en	84	81	71	56	sv:55	de:54	fr:52	**all:58**	ger:58
es	76	81	72	59	it:68	fr:65	ro:54	**rom:70**	ine:69
et	88	82	78	53	fi:65	hu:64	pl:63	**agl:75**	all:64
eu	78	73	66	**53**	**hu:45**	hi:41	et:39	sla:44	ine:43
fa	83	80	68	**55**	grc_1:47	he:46	sl:45	**sla:49**	all:44
fi	78	74	65	42	da:49	fi_1:47	et:45	**all:52**	agl:51
fi_1	73	69	65	45	fi:54	**la:49**	et:44	all:48	ine:47
fr	81	80	73	59	es:66	it:65	bg:57	**rom:67**	all:65
ga	78	73	70	**58**	**he:57**	id:53	ro:51	all:57	rom:56
got	82	77	73	58	**cu:67**	grc_1:66	la_1:54	sla:66	all:62
grc	72	68	59	45	grc_1:50	**got:49**	la:47	all:47	ine:47
grc_1	74	72	67	33	**got:62**	grc:52	la_1:49	sla:57	ine:56
he	84	81	76	**59**	id:55	es:54	ro:51	**rom:57**	all:57
hi	91	89	82	**60**	ta:55	hu:53	et:43	**agl:56**	all:47
hr	83	77	72	51	sl:57	cs:55	bg:54	**sla:61**	ine:60
hu	79	74	70	**62**	sv:54	bg:53	et:47	**sla:54**	all:53
id	82	79	70	**58**	**hr:57**	he:54	bg:48	sla:57	rom:53
it	87	86	79	64	es:74	fr:72	ro:59	**rom:76**	all:73
la	62	55	47	31	**grc_1:54**	cu:52	la_1:51	all:53	sla:52
la_1	72	69	58	43	**grc_1:56**	got:56	cu:54	sla:50	all:49
la_2	72	71	65	38	la:44	pl:44	hr:44	**sla:47**	ine:47
nl	73	71	68	53	de:52	pt:52	el:52	**ine:54**	all:53
no	87	84	72	42	sv:61	hr:61	bg:60	**ine:66**	ger:66
pl	87	83	78	62	sl:69	hr:67	bg:63	**sla:73**	all:69
pt	78	84	76	67	**it:69**	es:69	fr:66	all:68	rom:68
ro	76	66	62	53	it:59	id:58	es:56	**rom:62**	ine:61
sl	88	83	79	56	cs:70	hr:65	bg:62	**sla:75**	ine:69
sv	86	81	73	49	no:64	da:63	en:62	**ger:69**	ine:65
ta	80	66	63	**50**	hi:46	hu:44	eu:40	**agl:50**	all:48

Table 2: Unlabeled attachment score of delexicalized parsers on the UD 1.2 test data. Gold-standard tags were used in the "gold" column, and tags predicted by UDPipe everywhere else. The "gold", "lex" and "l20" columns are lexicalized. Parsers in "l20" are trained on 20 labeled sentences; **highlighted** figures indicate languages where the delexicalized parser did worse than "l20".

ing gold and predicted tags is not large (surprisingly, in three cases the predicted tags even outperform the gold standard). However, the UDPipe tagger was trained only on UD data and thus we observe a much larger drop in Romanian and Tamil—two tiny treebanks, too small to train a good tagger.

The "self/delex" column illustrates how much we lose by removing the lexical values of the words. Finally, we present scores of the three best-performing source languages, and two source language combinations. Again, genetically related languages tend to stick together and the scores could be used as an interesting language-typological metric, even if the correllation is less pronounced than with tagging. Sometimes the type of text plays a more important role. E.g. the treebanks from the PROIEL project (cu, got, grc_1 and la_1) work well together despite being from different language groups. They contain similar texts (Bible) and their annotation is harmonized to a greater extent than the rest of UD.

It is not surprising that the best possible source is usually a mix of languages (note that this cannot be attributed to larger training data, which is always limited to 5,000 sentences). Some of the UAS values look promising and certainly outperform unsupervised parsing. Yet there are two important factors that hold back excessive optimism. First, the results in Table 2 are based on a *supervised* tagger. Replacing it by the delexicalized tagger does not work for us. We do not show the scores in the table but they are generally under 20% and effectively useless. The number of tagging errors may not seem so disastrous, but their distribution is too random for the downstream parsing model to build upon the tags. Delexicalized tagging is reasonably good at distinguishing function words from content words but it often fails to tell apart nouns and verbs—a distinction that affects the entire structure of a dependency tree, whose root node is usually a verb.

The second factor lies in the "l20" column, which shows scores of a lexicalized parser trained on just 20 manually annotated sentences. (Zeman and Resnik, 2008) were able to show that their delexicalized transfer in their setup (with quite different data and parser from ours) was worth 1,546 manually annotated sentences. However, the learning curve of Malt Parser on UD data is much steeper in the beginning, making it harder for semi-supervised approaches to compete. Our results are rarely equivalent to more than 200 sentences of lexicalized data. In nine cases (highlighted red in the table) the delexicalized parser does not even outperform the "l20" result. This is certainly not good news for the delexicalized techniques, but there is a positive message, too: whenever some knowledge of the target language is available, use it. If a native speaker is available, ask him for help—even if he does not know anything about linguistics! The data you obtain may look ridiculously small, but they will probably get you further than expected.

6 Conclusion

We have investigated two techniques of cross-linguistic model transfer, known as delexicalized tagging and parsing. We evaluated them thoroughly on a common dataset, Universal Dependencies v1.2.

We have confirmed that models are more easily transferred between phylogenetically related languages—a hypothesis that is very natural, yet it could not be confirmed in the previous work, working with treebanks whose annotation style was not harmonized across languages (a famous finding of (McDonald et al., 2011) was that Danish was in fact the worst possible source language for Swedish).

We have also exposed the limits of the two methods. Each technique in isolation looks moderately promising, especially if only the intrinsic scores are considered. However, they fail terribly when used in combination. Since the learning curve of modern dependency parsers is quite steep, we argue that it is more advantageous to ask a native speaker to annotate just a small dataset, rather than trying to transfer models from other languages. We agree with (Yu et al., 2016) that such an approach does not scale well to tens or hundreds of languages, and a native speaker may not always be available, but it is a path that should not be ignored. Alternatively, one may want to employ Wiktionary crawling-based techniques such as (Sylak-Glassman et al., 2016) to acquire lexical knowledge about new languages.

Acknowledgments

The work was supported by the grants 15-10472S and 14-06548P of the Czech Science Foundation, and by the EU project H2020-ICT-2014-1-644402.

References

Željko Agić, Dirk Hovy, and Anders Søgaard. 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)*.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 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.

Christos Christodouloupoulos, 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.

Silviu Cucerzan and David Yarowsky. 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.

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*, pages 600–609, Portland, Oregon, USA, June.

Thành Duong, Steven Bird, Paul Cook, and Pavel Pecina. 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.

Victoria Fossum and Steven P. Abney. 2005. Automatically inducing a part-of-speech tagger by projecting from multiple source languages across aligned corpora. In Robert Dale, Kam-Fai Wong, Jian Su, and Oi Yee Kwong, 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.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2004. Bootstrapping parsers via syntactic projection across parallel texts. *Natural Language Engineering*, 1(1):1–15.

M. Paul Lewis, Gary F. Simons, and Charles D. Fennig, editors. 2016. *Ethnologue: Languages of the World, Nineteenth edition*. SIL International, Dallas, Texas.

Martin Majliš and Zdeněk Žabokrtský. 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.

Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-source transfer of delexicalized dependency parsers. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Joakim Nivre and Johan Hall. 2005. Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*, pages 13–95.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Portorož, Slovenia.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 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.

Rudolf Rosa and Zdeněk Žabokrtský. 2015. Klcpos3 - a language similarity measure for delexicalized parser transfer. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 243–249, Stroudsburg, PA, USA. Association for Computational Linguistics.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia.

John Sylak-Glassman, Christo Kirov, and David Yarowsky. 2016. Remote elicitation of inflectional paradigms to seed morphological analysis in low-resource languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, may.

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.

Zhiwei Yu, David Mareček, Zdeněk Žabokrtský, and Daniel Zeman. 2016. If you even don't have a bit of Bible: Learning delexicalized POS taggers. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 96–103, Portorož, Slovenia.

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