Multilingual Discriminative Shift-Reduce Phrase Structure Parsing for the SPMRL 2014 Shared Task

Benoit Crabbé
Alpage / INRIA
Université Paris Diderot
bcrabbe@univ-paris-diderot.fr

Djamé Seddah
Alpage / INRIA
Université Paris Sorbonne
djame.seddah@paris-sorbonne.fr

Abstract

This paper describes the design of a multilingual lexicalized discriminative shift reduce phrase structure based parser used to parse the SPMRL 2014 shared task data set. It reports the results of one of the first massively multilingual lexicalized phrase structure parser and shows that it behaves surprisingly well on a multilingual setting.

1 Introduction

This paper describes the design of a multilingual lexicalized discriminative shift-reduce phrase structure based parser used to parse the SPMRL 2014 shared task data set. Phrase structure parsing of a large range of morphologically rich languages is a relatively uncommon type of task: few parsers have actually been used for parsing wide range of languages at once[1]. The first one is that of Petrov et al. (2006) and has been used on the SPMRL 2013 dataset (Seddah et al., 2013b) by Björkelund et al. (2013) and Szántó and Farkas (2014) as a well as a baseline by the organizers. A second parser has been recently developed by Hall et al. (2014).

As a matter of fact, training dependency parsers on automatically converted dependency trees causes a loss of information and approximations during conversions can indeed have some impact on the resulting models (Simkó et al., 2014). Thus the paper investigates to which extent we can parse morphologically rich languages such as Arabic, German, French, Hungarian, Hebrew, Korean where the datasets are originally encoded as phrase structure trees and thus less susceptible to contain erroneous artifacts due to conversion errors.

Leaving aside the Hall et al. (2014)'s work, parsing experiments on the SPMRL dataset have been mainly carried out with the Berkeley parser (Björkelund et al., 2013; Szántó and Farkas, 2014). As this parsing model is generative, the inclusion of multiple morphological features typical to Morphologically rich languages into the model is not trivial, although it can be done with reasonable success, via a full rewrite of the original Petrov et al. (2006) lexical model, as shown in (Huang and Harper, 2011). One strategy otherwise often used is to simulate a feature model by extending the POS tagset with relevant morphological features (Crabbé and Candito, 2008; Dehdari et al., 2011; Szántó and Farkas, 2014).

The discriminative parsing model we use here (Crabbé, 2014) allows to take advantage of those features for parsing these languages reasonably accurately and significantly more efficiently[2] than the system engineered by Björkelund et al. (2013). The discriminative model allows to express several layers of backoff very naturally through features of different granularities. It should in principle be well suited for modelling morphologically rich languages with increased data sparsity issues. This paper thus describes a (very) preliminary work attempting to design a generic cross lingual parsing model into which we can plug features specific to each language.

[1] See (Tsarfaty et al., 2010; Tsarfaty et al., 2013) for references therein on previous works on Arabic, French, Hebrew, Czech, German among others.

[2] The parsing system used here has linear time complexity whereas (Björkelund et al., 2013) builds upon the polynomial parser of (Petrov et al., 2006).

The paper is organized as follows. In Section 2, we describe the lexicalized parser of (Crabbé, 2014). Since it is a lexicalized parser, it belongs to a family of algorithms that require the datasets to be annotated with heads. Since these annotations are not naturally present in the datasets, we provide a way to satisfy these requirements in Section 3. We describe there a method for assigning heads to the phrase structure treebanks automatically, by taking advantage of the alignment of phrase structure and dependency treebanks in the SPRML dataset. In Section 4, we describe a set of preliminary and still crude experiments designed to test the lexicalized framework in a multilingual setting. We report a very reasonable baseline on several languages without real feature engineering and then we quickly report results where the model attempts to take advantage of morphological and semi-supervised information for parsing morphologically rich languages.

2 Parsing System

The system used is an LR-inspired shift reduce phrase structure parser and has been described in (Crabbé, 2014). The parser relies on a binary head-markovized grammar which include all relevant morpho-syntactic feature information at the token level and latent syntactic information at the phrasal node level (eg. syntactic heads).

The parser itself is a shift reduce parser driven by an LR automaton. Given a configuration made of a stack and a queue $C = \langle S, Q \rangle$, the transition system is equipped with four groups of actions (Sagae and Lavie, 2006):

- A SHIFT action, which pushes the first element of the queue on the stack
- A set of reduce left(X) actions which pops the the two top elements of the stack and replaces them with the X element. X is a non terminal grammatical symbol. The X head element is its leftmost child.
- A set of Reduce Right(X) actions which pops the the two top elements of the stack and replaces them with the X element. The X head element is its righmost child.
- A set of Reduce Unary(X) actions which pops the top element from the stack and pushes an X element on top of the stack.

Given a current configuration C_i the parser has to to select a valid action for moving to the next configuration. Since this choice is naturally non-deterministic, each derivation $C_{0 \Rightarrow k} = C_0 \overset{a_0}{\Rightarrow} C_1 \Rightarrow \ldots \overset{a_{k-1}}{\Rightarrow} C_k$ is weighted by a function of the form:

$$W(C_{0 \Rightarrow k}) = \sum_{i=1}^{k-1} \mathbf{w} \cdot \mathbf{\Phi}(a_i, C_i)$$

The parser does not explore the full exponential space of derivations, instead it uses a beam, that is an approximate search heuristic restricting the number of active configurations at each time step. The best parse returned is the one that satisifies the equation:

$$\tilde{C} = \underset{C_{0 \Rightarrow 3n-1} \in \text{GEN}^K_{3n-1}(\mathcal{T})}{\text{argmax}} W(C_{0 \Rightarrow 3n-1})$$

where GEN^K_{3n-1} is the content of the beam after $3n-1$ derivation steps. In other words, the parser returns the highest weighted parse in the beam at the last derivation step.

The feature functions $\phi(a, C)$ have access to the top three elements of the stack and to the queue tokens, that is to the symbol categories and to their head tokens (Figure 1). Like dependency parsers, the parser uses structured lexical tokens, that is tokens made of a word form, but also features associated with this token such as morphological features.

The weight vector is estimated by a global perceptron model with early update and weight averaging (Collins, 2002).

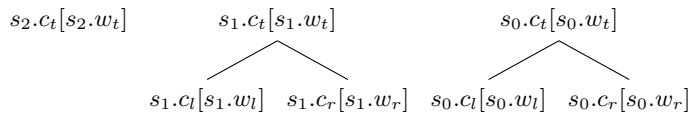

Figure 1: Information available to feature functions

3 Data set preprocessing

We describe here two essential preprocessing operations we run on the SPMRL data set. First, in order to make a lexicalized phrase structure parser operational over a large set of languages, we need to add head annotations. Second we normalized some morphological features recurring accross languages in order to facilitate their modelling across languages.

3.1 Head assignation procedure

In order to annotate phrase structure trees with heads, we take advantage of the aligned nature of the SPMRL datasets: for each language, dependency and constituency treebanks are aligned word-wise.

Since head annotation encodes projective dependencies in phrase structure trees, it seems natural to transfer the head information encoded in the dependency treebanks to their constituent counterparts.

However, the word aligned treebanks do not encode exactly the exact same structural information. We found that some dependency relations cannot be encoded directly in the corresponding phrase structure trees and vice-versa. This sometimes comes from two different treebanks that have been manually edited after automatic conversion such as the Hungarian treebank (Simkó et al., 2014), sometimes conversions add or remove information such that a node in a phrase structure tree has no matching node in the dependency tree with the same set of dependency relations. In other words a straightforward transfer is not possible in general.

We solved the problem in two steps. First, we ran a naive algorithm proceeding bottom up. The basis is that the leaf nodes are marked as heads of the unary rules introducing them. The recursion proceeds bottom up : for each rule $\alpha \rightarrow \beta_1[h_1] \dots \beta_n[h_n]$ of a phrase structure tree extract the set D of dependencies, (h_i, h') or (h', h_i) involving the headwords $h_1 \dots h_n$ from the dependency tree. If there is exacly one h_i dominating every $h_j \in D$ $(i \neq j)$ then h_i is set as head of the rule. Otherwise the procedure fails and it stops processing the tree. The naive algorithm is then run on a full treebank, and it collects the set R of rules together with their head index $h(r)$ for which the procedure succeeded.

The second step attempts to infer the heads for those trees where the naive algorithm fails. We processes again every tree bottom up. If the current rule r belongs to R then it is assigned $h(r)$ as head. In case $r \notin R$, we select the subset $R_k \subseteq R$ such that $R_k = \text{K-argmin}_{r' \in R} D(r, r')$. $D(r, r')$ is defined as the Levenshtein distance between the respective list of symbols of r and r'. Finally, the head index $h(r)$ is assigned to r by majority voting among the members of R_k[3]. In other words, the head assignation procedure can be seen as a variant of a $k-$nn procedure using the Levenshtein distance. In practice, we set K to 5.

We report in Table 1 the proportion of times the head assignation algorithm had to call the $k-$nn like guessing procedure for each SPMRL scenario. This metric roughly indicates how straightforward the head induction process is. As it can be seen this proportion is relatively low.

DATASET	ARABIC	BASQUE	FRENCH	GERMAN	HEBREW	HUNGARIAN	KOREAN	POLISH	SWEDISH
Gold (%)	0.9	6.7	0.01	2.3	0.06	0.6	0	0.01	0.1

Table 1: Proportion of guessed rules occurrences with respect to the total number of rule occurrences

[3]In case a nearest neighbor r' is a rule with a different arity than r, we initially designed a procedure aiming to enforce both rules to have the same arity using the Needleman-Wunsch alignment algorithm. However we observed that this case almost never occurs in practice, so we skip this detail here.

NORMALIZED	ARABIC	BASQUE	FRENCH	GERMAN	HEBREW	HUNGARIAN	KOREAN	POLISH	SWEDISH
subcat	subcat	AZP	subcat	-	-	SubPOS	tag	-	-
mood	mood	MDN	mood	mood	-	Mood	verb-type	-	verbform
gen	gen	-	g	gender	gen	-	-	gender	gender
num	num	NUM	n	number	num	Num	-	number	number
case	case	KAS	-	case	-	Cas	case-type	case	case
lem	lem	lem	lemma	lem	-	lemma	lem	lem	-

Table 2: Approximative feature normalization across languages

3.2 Cross lingual feature normalization

In order to design generic cross lingual models that can take into account some morphological information inherent to the task, we performed a superficial normalisation of the features encoded on the pre-terminals of the SPMRL phrase structure trees. We selected these features for normalisation since they are relatively recurrent accross languages and under the assumption that they could play a role for parsing purposes. We provide in table 2 the approximative mappings designed. The normalized features given in the left column are *mood, gen, num, case, lem,subcat* encoding respectively the mood, the gender, the number, the case, the lemma[4] and to some additional fine grained part of speech information found in several treebanks such as the definiteness of determiners. The values indicated in the table are the names of the features found in the actual datasets. These are tentative and approximate mappings that we inferred as well as we could from the documentation and the datasets themselves. When a treebank does not encode one of these normalised features, we added it to every token of this treebank with a constant dummy value. It should be noted that these mappings are preliminary and should be improved.

To ease comparison accross treebanks and accross models, we should at least normalize the tagsets. For instance, the Polish treebank encodes the mood within part of speech tags. The Korean tagset encodes far more information than the tagsets of other languages. Thus there is here plenty of room for improvement.

4 Experiments

The experiments are driven by the motivation of designing a generic parsing model common to every language expressed as a set of templates. Given such a model we would like to plug-in extensions specific to each language. In this shared task we mainly ran baseline experiments incorporating gradually along several models the information encoded in the data sets. We focused exclusively on the predicted morphology full size scenario.

We made five runs of experiments. The first experiment is a baseline model using a simplified set of templates described by (Zhu et al., 2013) for Chinese. The second adds some smoothing by attempting to improve the modelling of unknown words inspired by (Björkelund et al., 2013). The third and fourth models attempt to incorporate morphological information directly into the parsing model. While the fifth run attempts to address sparsity issues by incorporating clusters into the model.

The experiments use the implementation of the algorithm described by (Crabbé, 2014) which has been improved for efficiency and robustness. The parser beam is set to 8 and the perceptron is trained with an early update for 25 iterations. The final parsing model is not systematically averaged after 25 epochs. It is averaged from an epoch n such that the trainer score on the development set is maximal over all other epochs. Times are reported as average processing times in seconds per sentence over the development set. Times have been computed on a 2.4 ghz Intel iCore 7.

4.1 Baseline experiments

The first experiment uses a set of templates described in Figure 2 which is similar to those of Zhu et al. (2013). The templates can be read as follows: before the dot s_i and q_i denote respectively the position

[4]Note that the lemma feature used in the Arabic data set actually encodes the vocalized version of a token. Lemmas were not provided as part of the original treebank.

$s_0t.w_c$ & $s_0t.c$	$s_0t.w_f$ & $s_1t.w_f$	$s_0t.c$ & $s_1t.c$ & $s_2t.c$	$s_0t.c$ & $q_2.w_c$ & $q_3.w_c$
$s_0t.w_f$ & $s_0t.c$	$s_0t.w_f$ & $s_1t.c$	$s_0t.w_f$ & $s_1t.c$ & $s_2t.c$	$s_0t.c$ & $q_2.w_f$ & $q_3.w_c$
$s_1t.w_c$ & $s_1t.c$	$s_0t.c$ & $s_1t.w_f$	$s_0t.c$ & $s_1t.w_f$ & $q_0.w_c$	$s_0t.c$ & $q_2.w_c$ & $q_3.w_f$
$s_1t.w_f$ & $s_1t.c$	$s_0t.c$ & $s_1t.c$	$s_0t.c$ & $s_1t.c$ & $s_2t.w_f$	$s_0t.c$ & $s_0r.c$ & $s_1t.c$
$s_2t.w_c$ & $s_2t.c$	$s_0t.w_f$ & $q_0.w_f$	$s_0t.c$ & $s_1t.c$ & $q_0.w_c$	$s_0t.c$ & $s_0r.c$ & $s_1t.w_f$
$s_2t.w_c$ & $s_2t.c$	$s_0t.c$ & $q_0.w_f$	$s_0t.w_f$ & $s_1t.c$ & $q_0.w_c$	$s_0t.w$ & $s_0r.c$ & $s_1t.w_f$
$q_0.w_c$ & $q_0.w_f$	$s_0t.c$ & $q_0.w_c$	$s_0t.c$ & $s_1t.w_f$ & $q_0.w_c$	$s_0t.c$ & $s_0l.w_f$ & $s_1t.c$
$q_1.w_c$ & $q_1.w_f$	$q_0.w_f$ & $q_1.w_f$	$s_0t.c$ & $s_1t.c$ & $q_0.w_f$	$s_0t.c$ & $s_0l.c$ & $s_1t.w_f$
$q_2.w_c$ & $q_2.w_f$	$q_0.w_f$ & $q_1.w_c$	$s_0t.c$ & $q_0.w_c$ & $q_1.w_c$	$s_0t.c$ & $s_0l.c$ & $s_1t.c$
$q_3.w_c$ & $q_3.w_f$	$q_0.w_c$ & $q_1.w_c$	$s_0t.c$ & $q_0.w_f$ & $q_1.w_c$	
$s_0l.w_f$ & $s_0l.c$	$s_1t.w_f$ & $q_0.w_f$	$s_0t.c$ & $q_0.w_c$ & $q_1.w_f$	
$s_0r.w_f$ & $s_0r.c$	$s_1t.w_f$ & $q_0.w_c$	$s_0t.c$ & $q_1.w_c$ & $q_2.w_c$	
$s_1l.w_f$ & $s_1l.c$	$s_1t.c$ & $q_0.w_f$	$s_0t.c$ & $q_1.w_f$ & $q_2.w_c$	
$s_1r.w_f$ & $s_1r.c$	$s_1t.c$ & $q_0.w_c$	$s_0t.c$ & $q_1.w_c$ & $q_2.w_f$	

Figure 2: Baseline templates

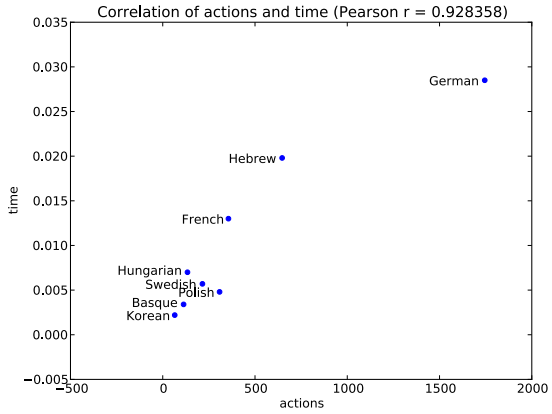

Figure 3: Correlation of times and number of actions

in the stack and in the queue of the adressed node as described in Figure 1. t, l, r denote the top, left and right nodes of the local trees in the stack. After the dot w_c, w_f denote a word category (tag) and a word form, while c is a constituent category.

This experiment involves two runs differing only on the management of low frequency words. The first run does not perform any explicit lexical smoothing while the second replaces word forms with a low frequency ($f \leq 2$) in the training set by its predicted part of speech. Results are reported in table 4

It can be seen that the effect of naive smoothing of low frequency words ranges from almost no effect to mild improvements on parsing accurracy accross languages.

More importantly the first run illustrates two key properties of the parser. First it is robust: parsing coverage reaches 100% for every language. Second it is efficient: parsing times are close to 0.01 second per sentence for all languages. There are however quite a few problematic cases such as German where our preprocessing methods turned out to be too simplistic: the closure of unary rules induces a very large number of non terminal symbols and therefore a huge number of parsing actions. This entails a significant loss of efficiency and it is very likely to be detrimental for accurracy too altough we did not try any alternative grammatical representation for this shared task.

We observe that the parsing complexity is a function of the form $\mathcal{O}(K|\mathcal{A}|n)$ where K is the size of the beam, $|\mathcal{A}|$, the number of actions and n the length of the sentence. Although it is common to express complexity as a function of n, hence $\mathcal{O}(n)$. And indeed the parsing time is linear in practice as a function of H. Across languages, we observe however from Figure 3 that the constant $|\mathcal{A}|$ is the most correlated factor to parsing times $r = 0.92$ whereas the correlation of parsing times with sentence length is almost non existent ($r =?$). This is caused by the fact that for most languages $|\mathcal{A}| >> \bar{n}$. One may wonder why the parser has to manage such a huge set of actions for some languages. Let $|N|$ be the number of non terminals in the grammar, then we have that $|\mathcal{A}| = 3|N| + 1$: three families of reductions and 1

shift. The sometimes large size of $|N|$ is mainly due to a large number of unary rules that are encoded as non terminal symbols by our treebank preprocessing function. There are several ways to improve the situation in the future, most notably by structuring the set of actions at inference time. This could improve speed drastically for languages with a large number of symbols and maybe also the accurracy.

4.2 Extended experiments

The extended experiment aims at introducing additional source of information in the parsing model. Since the SPMRL data set provides morphological features for all languages we included them in the parsing model. We did it in two steps. First we added morphological features to the baseline model. Second we also added lemmatized forms to the morphological parsing model by duplicating templates accessing word forms with templates accessing lemmas.

$s_0t.w_s$ & $s_0t.c$	$q_3.w_m$ & $q_0.w_f$	$s_0t.w_f$ & $s_1t.c$ & $q_0.w_s$
$s_0t.w_m$ & $s_0t.c$	$q_3.w_{cs}$ & $q_0.w_f$	$s_0t.w_f$ & $s_1t.w_f$ & $q_0.w_s$
$s_0t.w_{cs}$ & $s_0t.c$	$s_0t.c$ & $q_0.w_s$	$s_0t.c$ & $q_0.w_s$ & $q_1.w_s$
$s_1t.w_s$ & $s_0t.c$	$s_0t.c$ & $q_0.w_m$	$s_0t.c$ & $q_0.w_f$ & $q_1.w_s$
$s_1t.w_m$ & $s_0t.c$	$s_0t.c$ & $q_0.w_{cs}$	$s_0t.c$ & $q_0.w_s$ & $q_1.w_f$
$s_1t.w_{cs}$ & $s_0t.c$	$q_0.w_f$ & $q_1.w_s$	$s_0t.c$ & $q_1.w_s$ & $q_2.w_s$
$s_2t.w_s$ & $s_0t.c$	$q_0.w_f$ & $q_1.w_m$	$s_0t.c$ & $q_1.w_f$ & $q_2.w_s$
$s_2t.w_m$ & $s_0t.c$	$q_0.w_f$ & $q_1.w_{cs}$	$s_0t.c$ & $q_1.w_s$ & $q_2.w_f$
$s_2t.w_{cs}$ & $s_0t.c$	$q_0.w_c$ & $q_1.w_s$	$s_0t.c$ & $q_1.w_s$ & $q_2.w_s$
$q_0.w_s$ & $q_0.w_f$	$q_0.w_c$ & $q_1.w_m$	$s_0t.c$ & $q_2.w_f$ & $q_3.w_s$
$q_0.w_m$ & $q_0.w_f$	$q_0.w_c$ & $q_1.w_{cs}$	$s_0t.c$ & $e(s_0t.agr, s_1t.agr)$ & $s_1t.c$
$q_0.w_{cs}$ & $q_0.w_f$	$s_1t.w_c$ & $q_0.w_s$	$s_0t.c$ & $e(s_0t.num, s_1t.num)$ & $s_1t.c$
$q_1.w_s$ & $q_0.w_f$	$s_1t.w_c$ & $q_0.w_m$	$s_0t.c$ & $e(s_0t.gen, s_1t.gen)$ & $s_1t.c$
$q_1.w_m$ & $q_0.w_f$	$s_1t.w_c$ & $q_0.w_{cs}$	$s_0t.c$ & $e(s_0t.agr, q_0.agr)$ & $q_1.w_c$
$q_1.w_{cs}$ & $q_0.w_f$	$s_1t.c$ & $q_0.w_s$	$s_0t.c$ & $e(s_0t.gen, q_0.gen)$ & $q_1.w_c$
$q_2.w_s$ & $q_0.w_f$	$s_1t.c$ & $q_0.w_m$	$s_0t.c$ & $e(s_0t.num, q_0.num)$ & $q_1.w_c$
$q_2.w_m$ & $q_0.w_f$	$s_1t.c$ & $q_0.w_{cs}$	$s_0t.c$ & $e(s_0t.agr, q_1.agr)$ & $q_1.w_c$
$q_2.w_{cs}$ & $q_0.w_f$	$s_0t.c$ & $s_1t.w_f$ & $q_0.w_s$	$s_0t.c$ & $e(s_0t.num, q_0.num)$ & $q_1.w_c$
$q_3.w_s$ & $q_0.w_f$	$s_0t.c$ & $s_1t.c$ & $q_0.w_s$	

Table 3: Additional morphological related templates

We added morphological information from the normalized representation described in table 2. We add templates with access to subcat, mood, case, gender, number using the following notations at the right of the dot : $w_s, w_m, w_{cs}, w_g, w_n$. For modelling gender, number and agreement we additionally use the function $e(\cdot, \cdot)$ to denote an equality function returning true if the values of both its arguments are equal.

Thus the model extended with morphology is made of the baseline model templates augmented with the morphological templates described in Table 3. We also designed a last model where we duplicated every template involving a word form (suffixed by $.w_f$ in our notation) by a templates involving word lemmas. The result on the development set are reported in table 4.

From these results we observe that situations are different accross languages. For some languages, additional morphological information helps the parser (e.g. Basque, Hungarian, Polish, Swedish) for some others it is detrimental. These first observations should be interpreted with extreme care since the datasets have very different properties. An inspection of the learning curves revealed that these raw numbers hide very different situations. For instance, for some languages such as French and notably German, the learner seems to fit the training data poorly. This suggests that the current model lacks some critical information to actually account for these languages: adding more features does not help to improve accurracy. For other languages such as Swedish the model overfits the training data in the baseline setup: for such languages the additional features generally provided some mild to significant improvements.

More generally adding more and more features is likely to create overfitting situations and we actually do not know which ones help to get better generalisations. In further work, we definitely need to set up a more solid feature selection procedure such as described by (Ballesteros, 2013) or by using regularized models (L1 regularized models) in order to avoid the burden of manually engineering such features. In the current state of the parser, we are quite sure that our results remain far from being optimal.

(Models)	ARABIC	BASQUE	FRENCH	GERMAN	HEBREW	HUNGARIAN	KOREAN	POLISH	SWEDISH
Unsmoothed	78.91	76.05	**80.14**	**81.73**	86.78	83.57	79.18	91.66	76.27
Smoothed	79.35	76.02	79.83	81.23	86.89	83.96	**79.66**	92.12	77.66
Time(unsmoothed)	*0.0162*	*0.0034*	*0.013*	*0.0285*	*0.0198*	*0.007*	*0.0022*	*0.0048*	*0.0057*
Morph	79.23	83.82	79.47	76.09	**87.18**	**86.23**	79.51	92.93	**77.93**
Lemmas	**79.43**	**84.13**	79.93	75.28	86.93	85.96	79.63	**93.33**	77.53
Time(morph)	*0.0162*	*0.0056*	*0.0189*	*0.0297*	*0.0273*	*0.0102*	*0.0038*	*0.0074*	*0.008*

Table 4: Baseline (up) and Morphologically-informed Models Results on the Dev Set. (down)
Models in Bold were used for submitting fully supervised run (run0 in Table 5)

4.3 Semi-supervised experiments

We took advantage of the unlabeled data set released by the organizers to generate Brown clusters (Brown et al., 1992) using Liang (2005) implementation. We generate 1,000 word clusters from both the unlabeled and training data sets, for words appearing at least 100 times (except for Korean and Swedish where the threshold was set to 60 given the relatively small size of these data set – resp. 40 and 24 millions tokens, compared to above 100 millions for the others). As opposed to our previous works on semi-supervized parsing where we replaced all tokens with morphologically-enriched clusters in a PCFG-LA framework (Candito and Crabbé, 2009; Candito and Seddah, 2010; Seddah et al., 2013a), we decided to test the impact of the clusters as single features to be combined with other-morpho syntactic and lexical information.

We integrated them following three schemes: (run1) simple scheme where cluster features are added to the baseline feature template exposed in Fig. 2; (run2) a brute force scheme: the baseline feature template is extended with a replacement of all word (resp. pos tag) features by cluster features. Leading to a three time increase in feature size; (run3) same as run3 with lexical smoothing.

Results (Table 5) show a disappointing but recurring trend among the shared task participants (Seddah et al., 2014), using hard clusters fails to improve over a rich morphological feature model (run0). The only case where it brings an slight gain is for Korean using cluster features and lexical smoothing. Strangely, the run1 and run2 configurations perform roughly the same, while the generic cluster templates constantly under-perform with a large margin on Arabic, Basque, German, Hebrew, Hungarian and Korean. The only difference between the run2 and run3 models being the added small lexical smoothing for the latter, we believe that better performance would have been obtained on clusters built on lemmatized corpora (as shown by (Versley, 2014) on German and Swedish and by Candito and Seddah (2010) on French). By lack of time we could not test those configurations.

(Models)	ARABIC	BASQUE	FRENCH	GERMAN	HEBREW	HUNGARIAN	KOREAN	POLISH	SWEDISH
run0	**77.66**	**85.35**	**79.68**	**77.15**	**86.19**	**87.51**	79.35	**91.6**	**82.72**
run1	77.28	79.91	78.68	**76.56**	85.62	**86.62**	79.19	**90.90**	**81.92**
run2	74.25	75.9	78.39	74.75	83.53	82.48	61.57	90.01	80.56
run3	**77.36**	**80.12**	**79.13**	76.14	**85.71**	86.59	**79.50**	90.6	81.51

Table 5: Results of Submitted Models on the Test Set

5 Conclusion and perspectives

Our question for the shared task was to test to which extent approximative shift reduce lexicalized parsing scales up to the multilingual setting. To our knowledge, this is one of the first lexicalized parser to be run on a multilingual setting. This comes from two facts: (1) contrary to (Charniak, 2000) and (Collins, 2003) parsers our implementation does not hard code any language specific features and (2) we were able to generate head annotations for a wide range of languages using a systematic procedure which remains partly tied to the design of the SPMRL data sets.

From the current state of our system we observe that we can get a very efficient parser which is also reasonably accurate. For most languages we get comparable results to the baseline Berkeley parser with relatively few efforts spent on feature engineering. There is however room for improving accurracy if we compare with the results reported by the IMS-WROCLAW team that engineered the Berkeley parser in

order to maximize its accurracy: their setup involves products of grammars and reranking among others. However we expect our parser to be at least an order of magnitude faster than theirs.

We highlight some problems observed and some possible solutions for achieving a truly cross-lingual system. The first problem is to improve the management of unary rules that triggered some difficult situations for some languages. When the parser needs to decide among several thousand of actions at each time step, it is not surprising that the results are getting worse.

Another important issue will be to design a more stable learning procedure. We believe that improving the modelling of rare events is a key issue to the accurate modelling of morphologically rich languages. In the future we plan to dedicate some specific efforts to regularizing the parsing model. We specifically plan to replace the current perceptron model with a large margin estimation procedure and we also plan to automate feature selection in order to reduce feature engineering efforts.

Although related to the modelling of rare events, the third issue is related to the use of external semi-supervised kind of information. We could not, in most cases, take advantage of them. Even though it was a bit early to tackle this issue at this stage of development of the system, taking advantage of unlabelled data will be one of our major research directions in the future.

References

Miguel Ballesteros. 2013. Effective morphological feature selection with maltoptimizer at the spmrl 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 53–60. Citeseer.

Anders Björkelund, Ozlem Cetinoglu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (Re)ranking meets morphosyntax: State-of-the-art results from the SPMRL 2013 shared task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 134–144, Seattle, WA.

Peter F. Brown, Vincent J. Della, Peter V. Desouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Marie Candito and Benoit Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proc. of IWPT'09*, Paris, France.

Marie Candito and Djamé Seddah. 2010. Parsing word clusters. In *Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 76–84, Los Angeles, CA, USA, June. Association for Computational Linguistics.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of the 1st Annual Meeting of the North American Chapter of the ACL (NAACL)*, Seattle.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *EMNLP-2002*.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(3).

Benoit Crabbé and Marie Candito. 2008. Expériences d'analyse syntaxique statistique du français. In *Actes de la 15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN'08)*, pages 45–54, Avignon, France.

Benoit Crabbé. 2014. An LR-inspired generalized lexicalized phrase structure parser. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 541–552, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Jon Dehdari, Lamia Tounsi, and Josef van Genabith. 2011. Morphological features for parsing morphologically-rich languages: A case of arabic. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 12–21, Dublin, Ireland, October. Association for Computational Linguistics.

David Hall, Greg Durrett, and Dan Klein. 2014. Less grammar, more features. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–237, Baltimore, Maryland, June. Association for Computational Linguistics.

Zhongqiang Huang and Mary P Harper. 2011. Feature-rich log-linear lexical model for latent variable pcfg grammars. In *IJCNLP*, pages 219–227.

Percy Liang. 2005. Semi-supervised learning for natural language. In *MIT Master's thesis*, Cambridge, USA.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL*, Sydney, Australia.

Kenji Sagae and Alon Lavie. 2006. A best-first probabilistic shift-reduce parser. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 691–698. Association for Computational Linguistics.

Djamé Seddah, Marie Candito, and Enrique Henestroza Anguiano. 2013a. A word clustering approach to domain adaptation: Robust parsing of source and target domains. *Journal of Logic and Computation*.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clergerie. 2013b. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, WA.

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho Choi, Matthieu Constant, Richárd Farkas, Iakes Goenaga, Koldo Gojenola, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Joakim Nivre, Adam Przepiorkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, Alina Wróblewska, and Eric Villemonte de la Clérgerie. 2014. Overview of the spmrl 2014 shared task on parsing morphologically rich languages. In *Notes of the SPMRL 2014 Shared Task on Parsing Morphologically-Rich Languages*, Dublin, Ireland.

Katalin Ilona Simkó, Veronika Vincze, Zsolt Szántó, and Richárd Farkas. 2014. An empirical evaluation of automatic conversion from constituency to dependency in hungarian. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1392–1401, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Zsolt Szántó and Richárd Farkas. 2014. Special techniques for constituent parsing of morphologically rich languages. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 135–144, Gothenburg, Sweden, April. Association for Computational Linguistics.

Reut Tsarfaty, Djame Seddah, Yoav Goldberg, Sandra Kübler, Marie Candito, Jennifer Foster, Yannick Versley, Ines Rehbein, and Lamia Tounsi. 2010. Statistical parsing for morphologically rich language (SPMRL): What, how and whither. In *Proceedings of the First workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL)*, Los Angeles, CA.

Reut Tsarfaty, Djamé Seddah, Sandra Kübler, and Joakim Nivre. 2013. Parsing morphologically rich languages: Introduction to the special issue. *Computational Linguistics*, 39(1):15–22.

Yannick Versley. 2014. Incorporating semi-supervised features into discontinuous easy-first constituent parsing. In *Notes of the SPMRL 2014 Shared Task on Parsing Morphologically-Rich Languages*, Dublin, Ireland.

Muhua Zhu, Yue Zhang, Wenliang Chen, Min Zhang, and Jingbo Zhu. 2013. Fast and accurate shift-reduce constituent parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 434–443, Sofia, Bulgaria, August. Association for Computational Linguistics.