Parsing word clusters

Marie Candito[⋆] **and Djamé Seddah**[⋆⋄]
⋆ Alpage (Université Paris 7/INRIA), 30 rue du château des rentiers 75013 Paris, France
⋄ Université Paris-Sorbonne, 28, rue Serpente, 75006 Paris, France

Abstract

We present and discuss experiments in statistical parsing of French, where terminal forms used during training and parsing are replaced by more general symbols, particularly clusters of words obtained through unsupervised linear clustering. We build on the work of Candito and Crabbé (2009) who proposed to use clusters built over slightly coarsened French inflected forms. We investigate the alternative method of building clusters over lemma/part-of-speech pairs, using a raw corpus automatically tagged and lemmatized. We find that both methods lead to comparable improvement over the baseline (we obtain F_1=86.20% and F_1=86.21% respectively, compared to a baseline of F_1=84.10%). Yet, when we replace gold lemma/POS pairs with their corresponding cluster, we obtain an upper bound (F_1=87.80) that suggests room for improvement for this technique, should tagging/lemmatisation performance increase for French.

We also analyze the improvement in performance for both techniques with respect to word frequency. We find that replacing word forms with clusters improves attachment performance for words that are originally either unknown or low-frequency, since these words are replaced by cluster symbols that tend to have higher frequencies. Furthermore, clustering also helps significantly for medium to high frequency words, suggesting that training on word clusters leads to better probability estimates for these words.

1 Introduction

Statistical parsing techniques have dramatically improved over the last 15 years, yet lexical data sparseness remains a critical problem. And the richer the morphology of a language, the sparser the treebank-driven lexicons will be for that language.

Koo et al. (2008) have proposed to use word clusters as features to improve graph-based statistical dependency parsing for English and Czech. Their clusters are obtained using unsupervised clustering, which makes it possible to use a raw corpus containing several million words. Candito and Crabbé (2009) applied clustering to generative constituency parsing for French. They use a *desinflection* step that removes some inflection marks from word forms and then replaces them with word clusters, resulting in a significant improvement in parsing performance. Clustering words seems useful as a way of addressing the lexical data sparseness problem, since counts on clusters are more reliable and lead to better probability estimates. Clustering also appears to address the mismatch of vocabularies between the original treebank and any external, potentially out-of-domain corpus: clusters operate as an intermediary between words from the treebank and words from the external corpus used to compute clusters. Furthermore, parsing word clusters instead of word forms augments the known vocabulary.

However, depending on the clustering method, clusters are either not very reliable or are available only for very frequent words. In order to parse word clusters one needs to determine which word clusters are reliable enough to be beneficial, so the tuning of parameters such as cluster granularity and cluster reliability becomes very important.

The aim of this paper is to give an in-depth study of the "parsing word clusters" technique. In particular, starting from the Candito and Crabbé (2009) experiments, we investigate the use of clustering lem-

Proceedings of the NAACL HLT 2010 First Workshop on Statistical Parsing of Morphologically-Rich Languages, pages 76–84,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

mas instead of *desinflected* forms. We also provide an analysis of the performance gains obtained with respect to word frequency (frequent words, rare words, unknown words).

In the next section, we describe the French treebank used as the basis for all of our experiments. We describe in section 3 the statistical parser used for training and testing. We then describe the desinflection process used prior to unsupervised clustering (section 4), and the Brown algorithm we use for unsupervised clustering (section). We describe our experiments and results in section 6, and provide a discussion in section 7. We then point out some related work and conclude in section 9.

2 French Treebank

For our experiments, we used the French Treebank (Abeillé et al., 2003), which contains 12531 sentences, 350931 tokens, from the newspaper *Le Monde*. We used the treebank instantiation (hereafter FTB-UC) as first described in (Candito and Crabbé, 2009), where :
(i) the rich original annotation containing morphological and functional information is mapped to a simpler phrase-structure treebank with a tagset of 28 part-of-speech tags, and no functional annotation
(ii) some compounds with regular syntax are broken down into phrases containing several simple words
(iii) the remaining sequences annotated as compound words in the FTB are merged into a single token, whose components are separated with an underscore

For all experiments in this paper (tagging and parsing) we used the same partition of the treebank as these authors : first 10% for test, next 10% for dev and the rest for training[1].

3 Berkeley Parser

We report here experiments using the Berkeley PCFG parser with latent annotations (Petrov et al., 2006), hereafter BKY, which is a constituent parser that has been proven to perform well for French (Crabbé and Candito, 2008; Seddah et al., 2009), though a little lower than a combination of a tagger plus the dependency-based MST parser (Candito et al., 2010). Though PCFG-style parsers operate on too narrow a domain of locality, splitting symbols according to structural and/or lexical properties is known to help parsing (Klein and Manning., 2003). Following (Matsuzaki et al., 2005), the BKY algorithm uses EM to estimate probabilities on symbols that are automatically augmented with latent annotations, a process which can be viewed as symbol splitting. It iteratively evaluates each such split and merges back the less beneficial ones. Crabbé and Candito (2008) show that some of the information carried by the latent annotations is lexical, since replacing words by their gold part-of-speech tag leads to worse results than the corresponding perfect tagging test, with words unchanged. This is a clear indication that lexical distinctions are used, and percolate up the parse tree via the latent annotations.

We now describe how the BKY software handles rare and unknown words, as this is pertinent to our discussion in section 6. $P(w|tag)$ is calculated using Bayes' rule, as $P(tag|w)P(w)/P(tag)$. Relative frequency estimates are used for words that are sufficiently frequent. For rare words (appearing less than 10 times in our settings), $P(tag|w)$ is smoothed using the proportion of tokens in the second half of the training set that were not seen in the first half, and that have this tag. For unknown words, words signatures are used: these are word classes determined by information such as the word suffix, whether the word is capitalized, whether it contains digits, etc. $P(w|tag)$ is estimated with $P(signature(w)|tag)$, and is also smoothed in the same way rare words are.

4 Morphological clustering

A first approach to word clustering is to cluster forms on a morphological basis. In the case of a relatively morphologically rich language such as French, this is an obvious way to reduce lexical sparseness caused by inflection.

(Candito and Crabbé, 2009) proposed the use of a *desinflection* method, without resorting to part-of-speech tagging. We propose an alternate method here, which uses lemmas and part-of-speech tags that are output by a tagger/lemmatizer. Because

[1] More precisely the partition is : first 1235 sentences for test, next 1235 sentences for development, and remaining 9881 sentences for training.

counts on lemmas are more reliable, clustering over lemmas presumably produces clusters that are more reliable than those produced by clustering over desinflected forms. However, this approach does create a constraint in which automatically tagged and lemmatized text is required as input to the parser, leading to the introduction of tagging errors.

Both morphological clustering methods make use of the Le*fff* lexicon (Sagot, 2010). Before we describe these two methods, we briefly give basic information on French inflectional morphology and on the Le*fff*.

4.1 French inflection and the Le*fff* lexicon

French nouns appear in singular and plural forms, and have an intrinsic gender. The number and gender of a noun determines the number and gender of determiners, adjectives, past participles that depend on it. Hence in the general case, past participles and adjectives have four different forms. The major inflectional variation appears for finite verbs that vary for tense, mood, person and number. A regular verb may correspond to more than 60 inflected forms if all tenses and mood are included. In practice, some forms occur very rarely, because some tense/mood pairs are rare, and further, in the case of newspaper text for instance, the first and second persons are also rare. So for instance in the FTB-UC, there are 33 different forms for the highly frequent verb and auxiliary *avoir* (*to have*), that appears 4557 times. The medium frequency verb *donner* (*to give*) occurs 155 times, under 15 different forms. In the whole treebank, there are 27130 unique word forms, corresponding to 17570 lemmas.

The Le*fff* is a freely available rich morphological and syntactic French lexicon (Sagot, 2010). It contains $110,477$ lemmas (simple and compounds) and $536,375$ inflected forms. The coverage on the FTB-UC is high : around 96% of the tokens, and $80,1$% of the types are present in the Le*fff* (leaving out punctuation and numeric tokens, and ignoring case differences).

4.2 Desinflection

The aim of the desinflection step is to reduce lexical data sparseness caused by inflection, without hurting parsability and without committing oneself as far as lexical ambiguity is concerned. The idea is to leave unchanged the parser's task in disambiguating part-of-speech tags. In that case, morphological clustering using lemmas is not an option, since lemma assignment presupposes POS disambiguation. Furthermore, useful information such as verb mood (which is needed to capture, for instance, that infinitive verbs have no overt subject or that participial clauses are sentence modifiers) is discarded during lemmatization, though it is encoded in the FTB with different projections for finite verbs (projecting sentences) versus non finite verbs (projecting VPpart or VPinf).

The intuition of Candito and Crabbé (2009) is that other inflectional markers in French (gender and number for determiners, adjectives, pronouns and nouns, or tense and person for verbs) are not crucial for inferring the correct phrase-structure projection for a given word. Consequently, they proposed to achieve morphological clustering by *desinflection*, namely by removing unneeded inflectional markers, identified using the Le*fff*. This lexicon-based technique can be viewed as an intermediate method between stemming and lemmatization.

The desinflection process is as follows: for a token t to *desinflect*, if it is known in the lexicon, then for each inflected lexical entry le of t, try to get a corresponding singular entry. If corresponding singular entries exist for all such le and all have the same form, then replace t by the corresponding form. For instance for wt=*entrées* (ambiguous between *entrances* and *entered*, fem, plural), the two lexical entries are *[entrées/N/fem/plu]* and *[entrées/V/fem/plu/part/past]*[2], each have a corresponding singular lexical entry, with form *entrée*.

The same process is used to map feminine forms to corresponding masculine forms. This allows one to change *mangée* (*eaten*, fem, sing) into *mangé* (*eaten*, masc, sing). But for the form *entrée*, ambiguous between N and Vpastpart entries, only the participle has a corresponding masculine entry (with form *entré*). In that case, in order to preserve the original part-of-speech ambiguity, *entrée* is not replaced by *entré*. Finite verb forms, when unambiguous with other parts-of-speech, are mapped to second person plural present indicative corresponding forms. This choice was made in order to avoid cre-

[2] This is just an example and not the real Lefff format.

Dev set	Overall	Overall (-punct)	Unseen (4.8)
POS acc	97.38	96.99	91.95
Lemma acc	98.20	97.93	92.52
Joint acc	96.35	95.81	87.16
Test set	Overall	Overall (-punct)	Unseen (4.62)
POS acc	97.68	97.34	90.52
Lemma acc	98.36	98.12	91.54
Joint acc	96.74	96.26	85.28

Table 1: MORFETTE performance on the FTB-UC dev and test sets (with and without punctuation)

ating ambiguity: the second person plural forms end with a very typical *-ez* suffix, and the resulting form is very unlikely ambiguous. For the first token of a sentence, if it is unknown in the lexicon, the algorithm tries to desinflect the corresponding lowercase form.

This desinflection process reduces the number of distinct tokens in the FTB-UC training set from 24110 to 18052.

4.3 Part-of-speech tagging and lemmatization

In order to assign morphological tags and lemmas to words we use a variation of the MORFETTE model described in (Chrupała et al., 2008). It is a sequence labeling model which combines the predictions of two classification models (one for morphological tagging and one for lemmatization) at decoding time, using a beam search.
While (Chrupała et al., 2008) use Maximum Entropy training to learn P_M and P_L, we use the MORFETTE models described in (Seddah et al., 2010), that are trained using the Averaged Sequence Perceptron algorithm (Freund and Schapire, 1999). The two classification models incorporate additional features calculated using the Le*fff* lexicon.

Table 1 shows detailed results on dev set and test set of the FTB-UC, when MORFETTE is trained on the FTB-UC training set. To the best of our knowledge the parts-of-speech tagging performance is state-of-the-art for French[3] and the lemmatization performance has no comparable results.

5 Unsupervised clustering

We use the Brown et al. (1992) hard clustering algorithm, which has proven useful for various NLP tasks such as dependency parsing (Koo et al., 2008) and named entity recognition (Liang, 2005). The algorithm to obtain C clusters is as follows: each of the C most frequent tokens of the corpus is assigned its own distinct cluster. For the $(C+1)^{th}$ most frequent token, create a $(C+1)^{th}$ cluster. Then for each pair among the $C+1$ resulting clusters, merge the pair that minimizes the loss in the likelihood of the corpus, according to a bigram language model defined on the clusters. Repeat this operation for the $(C+2)^{th}$ most frequent token, etc. The result is a hard clustering of words in the corpus into C distinct clusters, though the process can be continued to further merge pairs of clusters among the C clusters, ending with a single cluster for the entire vocabulary. A binary tree hierarchy of merges for the C clusters can be obtained by tracing the merging process, with each cluster identified by its path within this binary tree. Clusters can thus be used at various levels of granularity.

6 Experiments and results

6.1 Clustering

For the Brown clustering algorithm, we used Percy Liang's code[4], run on the *L'Est Républicain* corpus, a 125 million word journalistic corpus, freely available at CNRTL[5]. The corpus was first tokenized and segmented into sentences. For compound words, the 240 most frequent compounds of the FTB-UC were systematically recognized as one token. We tried out the two alternate morphological clustering processes described in section 4 as a preprocessing step before the unsupervised clustering algorithm was run on the *L'Est Républicain* corpus :
(i) word forms were replaced by corresponding desinflected form
(ii) word forms were replaced by a concatenation of the part-of-speech tag and lemma obtained with MORFETTE[6].

[3] A pure MAXENT based tagger is described in (Denis and Sagot, 2009), that also uses the Le*fff*, under the form of features for the known categories of a word in the lexicon. The authors report 97.70% of accuracy and 90.01% for unseen data.

[4] *http://www.eecs.berkeley.edu/ pliang/software*

[5] *http://www.cnrtl.fr/corpus/estrepublicain*

[6] Because these experiments were first run with a version of Morfette that was not yet optimized for lemmatization, we chose to overide the MORFETTE lemma when the Le*fff* lemma is available for a given form and part-of-speech tag pair supplied by Morfette. Morfette's current results (version 0.3.1) in

Name	Terminal symbols in training set	Vocabulary size in training set	Terminal symbols in dev/test sets
Baseline	wf	24110	wf
Dfl	Desinflected wf	18052	Desinflected wf
Dfl+Clust>X	$Cluster_1$(desinflected wf)	1773 ($X = 200$)	$Cluster_1$(desinflected wf)
GoldCatLemma	Gold POS+lemma	18654	Gold POS+lemma
AutoCatLemma	Gold POS+lemma	18654	Automatic POS+lemma
GoldCatLemma+Clust>X	$Cluster_2$(gold POS+lemma)	1298 ($X = 200$)	$Cluster_2$(gold POS+lemma)
AutoCatLemma+Clust>X	$Cluster_2$(gold POS+lemma)	1298 ($X = 200$)	$Cluster_2$(automatic POS+lemma)

Table 2: Types of terminal symbols used for training and parsing

In the first case we obtain clusters of desinflected forms, whereas in the second case we obtain clusters of tag+lemma pairs. Note that lemmas alone could be used, but as noted earlier, important syntactic information would be lost, particularly for verb mood. We did try using clusters of lemmas, coupled with a few suffixes to record the verb mood, but this resulted in more or less the same performance as clusters of tag+lemma pairs.

6.2 Berkeley parser settings

For Bky we used Slav Petrov's code, adapted for French by Crabbé and Candito (2008) by modifying the suffixes used to classify unknown words. We use the partition between training, development and test sets introduced in section 2. Note though that the Bky algorithm itself uses two sets of sentences at training: a learning set and a smaller validation set for tuning model hyperparameters. In all experiments in this paper, we used 2% of the training set as as a validation set, and 98% as a learning set. This differs from (Candito and Crabbé, 2009), where the dev set was used as a validation set.

6.3 Experiments

We then tested several settings differing only in the terminal symbols used in the training set, and in the dev and test sets. We list these settings in table 2. For the settings involving unsupervised linear clustering:

Dfl+Clust>X: Each desinflected form df is replaced by $Cluster_1(df)$: if df occurred more than X times in the *L'Est Républicain* corpus, it is replaced by its cluster id, otherwise, a special cluster UNKC is used. Further, a _c suffix is added if the desinflected form starts with a capital letter, and additional features are appended, capturing whether the form is all digits, ends with *ant*, or *r*, or *ez* (cf. this is the ending of the desinflected forms of unambiguous finite verbs). (Candito and Crabbé, 2009) showed that these additional features are needed because clusters are noisy: linear context clustering sometimes groups together items that belong to different parts-of-speech.

GoldCatLemma+Clust>X: The terminal form used is the gold part-of-speech concatenated to the cluster id of the gold POS+lemma, or UNKC if that pair did not occur more than X times in the *L'Est Républicain* corpus.

AutocatLemma+Clust>X: For the training set, the same setting as GoldCatlemma+Clust>X is used. But for the dev and test sets, predicted parts-of-speech and lemmas are used, as output by the Morfette tagger/lemmatizer: the terminal form is the predicted part-of-speech concatenated with the cluster id of the predicted POS+lemma, or UNKC if that pair was not frequent enough.

For the Clust>X experiments, we report results with $X = 200$. We have found empirically that varying X between 20 and 700 has very little effect on performance gains, both for clustering of desinflected forms and clustering of tag+lemma pairs. Also, all results are with a maximum number of clusters set to 1000, and we found that limiting the number of clusters (by taking only a prefix of the cluster bit string) degrades results.

6.4 Evaluation metrics

We evaluate parsing performance using labeled F-Measure (combining labeled precision and labeled

lemmatization renders this step obsolete.

Dev set				
Terminal symbols	F_1<40	F_1	UAS	Tagging Acc.
Baseline	86.06	**83.81**	**89.23**	96.44
Dfl	86.65	84.67 (+0.86)	89.86	96.52
Dfl+Clust>200	87.57	**85.53** (+1.72)	**90.68**	96.47
AutoCatLemma	86.77	84.52 (+0.71)	89.97	96.25
AutoCatLemma+Clust>200	87.53	**85.19** (+1.38)	**90.39**	96.78
GoldCatLemma	87.74	85.53 (+1.72)	91.42	98.49
GoldCatLemma+Clust>200	88.83	86.52 (+2.71)	92.11	99.46

Test set				
Terminal symbols	F_1<40	F_1	UAS	Tagging Acc.
Baseline	86.16	**84.10**	**89.57**	96.97
Dfl	87.13	85.07 (+0.93)	90.45	97.08
Dfl+Clust>200	88.22	**86.21** (+2.11)	**90.96**	96.98
AutoCatLemma	86.85	84.83 (+0.73)	90.30	96.58
AutoCatLemma+Clust>200	87.99	**86.20** (+2.10)	**91.22**	97.11
GoldCatLemma	88.16	85.90 (+1.80)	91.52	98.54
GoldCatLemma+Clust>200	89.93	87.80 (+3.70)	92.83	99.41

Table 3: Parsing performance on the dev set/test set when training and parsing make use of clustered terminal symbols. F_1<40 is the F-Measure combining labeled precision and labeled recall for sentences of less than 40 words. All other metrics are for all sentences of the dev set/test set. UAS = Unlabeled attachement score of converted constituency trees into surface dependency trees. All metrics ignore punctuation tokens.

recall) both for sentences of less than 40 words, and for all sentences[7]. We also use the unlabeled attachment score (UAS), obtained when converting the constituency trees output by the Bky parsers into surface dependency trees, using the conversion procedure and software of (Candito et al., 2010)[8]. Punctuation tokens are ignored in all metrics.

7 Discussion

Results are shown in table 3. Our hope was that using lemmatization would improve overall accuracy of unsupervised clustering, hence leading to better parsing performance. However, results using both methods are comparable.

Table 3 shows that both morphological clustering techniques (Dfl and AutoCatLemma) slightly improve performance ($+0.97$ and $+0.73$ F_1 over the baseline for the test set)[9]. In the case of AutoCatLemma, morphological ambiguity is totally absent in training set: each terminal symbol is the gold POS+lemma pair, and hence appears with a unique part-of-speech in the whole training set. But at parsing time, the terminal symbols are the POS+lemma pairs predicted by Morfette, which are wrong for approximately 3% of the tokens. So when comparing the impact on parsing of the two morphological clustering techniques, it seems that the advantage of lemmatization (a sounder morphological clustering compared to the desinflection process) is counterbalanced by tagging errors that lead to wrong POS+lemma pairs. Indeed, it can be verified that

[7] Note that often for statistical constituent parsing results are given for sentences of less than 40 words, whereas for dependency parsing, there is no such limitation. The experiment Dfl and Dfl+Clust>200 are reproduced from the previous work (Candito and Crabbé, 2009). More precisely, this previous work reports $F_1 = 88.29$ on the test set, but for sentences ≤ 40 words, for a Dfl+Clust>20 experiment, and as previously mentioned, the dev set was used as validation set for the Bky algorithm. We report now $F_1 = 88.22$ for the same less-than-40-words sentences, leaving dev set unused at training time.

[8] The conversion uses head propagation rules to find the head on the right-hand side of the CFG rules, first proposed for English in (Magerman, 1995). Hence the process is highly sensitive to part-of-speech tags.

[9] We have computed p-values for pairs of results, using Dan Bikel's statistical significance tester for evalb output (http://www.cis.upenn.edu/ dbikel/software.html). All experiments have a p-value < 0.05 both for recall and precision when compared to the baseline. The differences between Dfl and AutoCatLemma, and between Dfl+Clust>200 and AutoCatLemma+Clust>200 are not statistically significant ($p - value > 0.2$). The gain obtained by adding the unsupervised clustering is clearly significant ($p - value > 0.005$), both when comparing AutoCatLemma and AutoCatLemma+Clust>200, and Dfl and Dfl+Clust>200.

		Baseline		Dfl+Clust>200		AutoCatLemma+Clust>200	
Frequency range in original training set	#tokens in dev set	UAS	Tagging	UAS	Tagging	UAS	Tagging
any	31733 (100%)	89.23	96.43	90.68	96.45	90.39	96.78
0 (original unseen)	1892 (5.96%)	84.78	82.56	88.79	89.22	88.64	91.17
$0 < x \leq 5$	3376 (10.64%)	86.49	94.52	88.68	93.13	88.33	95.41
$5 < x \leq 10$	1906 (6.01%)	90.35	96.59	91.50	95.02	91.55	96.12
$10 < x \leq 20$	2248 (7.08%)	89.55	96.71	91.37	95.42	90.57	95.91
$20 < x \leq 50$	3395 (10.70%)	91.87	96.35	92.40	95.96	91.72	95.94
$x \geq 50$	18916 (59.61%)	89.53	98.12	90.75 (+1.22)	98.12	90.56 (+1.03)	97.91

Table 4: Tagging accuracy and UAS scores for words in the dev set, grouped by ranges of frequencies in the **original** training set.

when parsing the gold POS+lemma pairs (the non realistic GoldCatLemma setting[10]), performance is greatly improved ($+1.80$ F_1 over the baseline).

Replacing morphological clusters by their corresponding unsupervised clusters leads to a further improvement, both for F_1 score and for UAS. But here again, using desinflection or tagging+lemmatisation leads to more or less the same improvement. But while the former method is unlikely improvable, the latter method might reveal more effective if the performance of the tagging/lemmatisation phase improves. The GoldCatLemma+Clust>200 experiment gives the upper bound performance : it leads to a $+3.70 F_1$ increase over the baseline, for the test set. In that case, the terminal symbols are made of the perfect POS plus the cluster of the perfect POS+lemma pair. Very few such terminal symbols are unseen in training set, and all are unambiguous with respect to part-of-speech (hence the 99.46% tagging accuracy).

In order to better understand the causes of improvement, we have broken down the tagging accuracy scores and the UAS scores according to various ranges of word frequencies. For word forms in the dev set that occur x times *in the original training set*, for x in a certain range, we look at how many are correctly tagged by the parsers, and how many receive the correct head when constituents are converted into surface dependencies.

The results are shown in table 4. Unseen words and rare words are much better handled (about $+4$ points for UAS for unseen words, and $+2$ points for forms appearing less than 5 times). This is simply obtained because the majority of original rare or unknowns are replaced by terminal symbols (either a cluster id or the UNKC token, plus suffixes) that are shared by many forms in the treebank, leading to higher counts. This can be verified by analyzing tagging accuracies and UAS scores for various frequency ranges for the *modified* terminal symbols : the symbols that replace the word forms in the training set for Dfl+Clust>X and AutoCatLemma+Clust>X experiments. This is shown in table 5. It can be seen that the majority of the tokens have now high-frequency. For instance for the Dfl+Clust>200 experiment, there are only 0.09% terminal symbols in the dev set that are unseen in training set, and 92.62% appear more than 50 times. The parsers do not perform very well on low-frequency modified terminal symbols, but they are so few that it has little impact on the overall performance.

Hence, in our parsing word clusters experiments, there are almost no real unseen anymore, there are only terminal symbols made of a cluster id or UNKC (plus suffixes). More precisely, for instance about 30% of the original unseen in the dev set, are replaced by a UNKC* symbol, which means that 70% are replaced by a cluster-based symbol and are thus “connected” to the known vocabulary.

Interestingly, the improvement in performance is also evident for words with high frequency in the original treebank: for the forms appearing more than 50 times in the original training set, the UAS increases by $+1.22$ with Dfl+Clust>200 and $+1.03$ with AutoCatLemma+Clust>200 (table 4). This means that despite any imperfections in the

[10] The GoldCatLemma experiment leads to high tagging accuracy, though not perfect, because of POS+lemma pairs present in dev/test sets but missing in the training set.

	DFL+CLUST>200			AUTOCATLEMMA+CLUST>200		
FREQUENCY RANGE in modified training set	percentage of dev set	UAS	Tagging	percentage of dev set	UAS	Tagging
any	100	90.68	96.45	100	90.39	96.78
0 (effective unseen)	0.09	86.21	58.62	0.08	84.00	40.00
$0 < x \leq 5$	0.45	88.19	86.81	0.32	70.30	70.30
$5 < x \leq 10$	0.64	90.69	91.18	0.37	92.24	79.31
$10 < x \leq 20$	1.31	90.12	94.22	0.87	89.53	88.09
$20 < x \leq 50$	4.88	88.64	92.19	3.30	86.44	92.65
$x \geq 50$	92.62	90.81	96.83	95.07	90.60	97.21
replaced by UNKC*	8.58	90.64	88.10	8.73	89.67	90.07

Table 5: Tagging accuracy and UAS scores for modified terminal symbols in the dev set, grouped by ranges of frequencies in the modified training sets. The "replaced by UNKC*" line corresponds to the case where the desinflected form or the POS+lemma pair does not appear more than 200 times in the *L'est Républicain* corpus.

unsupervised Brown clustering, which uses very local information, the higher counts lead to better estimates even for high-frequency words.

8 Related work

We have already cited the previous work of Koo et al. (2008) which has directly inspired ours. Sagae and Gordon (2009) explores the use of syntactic clustering to improve transition-based dependency parsing for English : using an available 30 million word corpus parsed with a constituency parser, words are represented as vectors of paths within the obtained constituency parses. Words are then clustered using a similarity metric between vectors of syntactic paths. The clusters are used as features to help a transition-based dependency parser. Note that the word representation for clustering is more complex (paths in parse trees), thus these authors have to cluster a smaller vocabulary : the top 5000 most frequent words are clustered.

Agirre et al. (2008) use the same approach of replacing words by more general symbols, but these symbols are semantic classes. They test various methods to assign semantic classes (gold semantic class, most-frequent sense in sense-tagged data, or a fully unsupervised sense tagger). Though the method is very appealing, the reported improvement in parsing is rather small, especially for the fully unsupervised method.

Versley and Rehbein (2009) cluster words according to linear context features, and use the clusters as features to boost discriminative German parsing for unknown words. Another approach to augment the known vocabulary for a generative probabilistic parser is the one pursued in (Goldberg et al., 2009). Within a plain PCFG, the lexical probabilities for words that are rare or absent in the treebank are taken from an external lexical probability distribution, estimated using a lexicon and the Baulm-Welch training of an HMM tagger. This is proven useful to better parse Hebrew.

9 Conclusion and future work

We have provided a thorough study of the results of parsing word clusters for French. We showed that the clustering improves performance both for unseen and rare words and for medium- to high-frequency words. For French, preprocessing words with desinflection or with tagging+lemmatisation lead to comparable results. However, the method using POS tagging is expected to yield higher performance should a better tagger become available in the future.

One avenue for further improvement is to use a clustering technique that makes explicit use of syntactic or semantic similarity, instead of simple linear context sharing. While the Brown clustering algorithm can be run on large raw corpus, it uses extremely local information (bigrams). The resulting clusters are thus necessarily noisy, and semantic or syntactic clustering would certainly be more appropriate. Since resource-based semantic clustering is difficult for French due to a lack of resources, clustering based on distributional syntactic similarity is a worthwhile technique to investigate in the future.

Acknowledgments

This work was supported by the ANR Sequoia (ANR-08-EMER-013). We are grateful to our anonymous reviewers for their comments and to Grzegorz Chrupala for helping us with Morfette.

References

Anne Abeillé, Lionel Clément, and François Toussenel, 2003. *Building a Treebank for French*. Kluwer, Dordrecht.

Eneko Agirre, Timothy Baldwin, and David Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of ACL-08: HLT*, pages 317–325, Columbus, Ohio, June. Association for Computational Linguistics.

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 Benoît Crabbé. 2009. Improving generative statistical parsing with semi-supervised word clustering. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 138–141, Paris, France, October. Association for Computational Linguistics.

Marie Candito, Benoit Crabbé, and Pascal Denis. 2010. Statistical french dependency parsing : Treebank conversion and first results. In *Proceedings of LREC'2010*, Valletta, Malta.

Grzegorz Chrupała, Georgiana Dinu, and Josef van Genabith. 2008. Learning morphology with morfette. In *In Proceedings of LREC 2008*, Marrakech, Morocco. ELDA/ELRA.

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.

Pascal Denis and Benoît Sagot. 2009. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art pos tagging with less human effort. In *Proc. of PACLIC*, Hong Kong, China.

Yoav Freund and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296.

Yoav Goldberg, Reut Tsarfaty, Meni Adler, and Michael Elhadad. 2009. Enhancing unlexicalized parsing performance using a wide coverage lexicon, fuzzy tag-set mapping, and EM-HMM-based lexical probabilities. In *Proc. of EACL-09*, pages 327–335, Athens, Greece.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08*, pages 595–603, Columbus, USA.

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

D.M. Magerman. 1995. Statistical decision-tree models for parsing. In *Proc. of ACL'95*, pages 276–283, Morristown, NJ, USA.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 75–82.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proc. of ACL-06*, Sydney, Australia.

Kenji Sagae and Andrew S. Gordon. 2009. Clustering words by syntactic similarity improves dependency parsing of predicate-argument structures. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 192–201, Paris, France, October. Association for Computational Linguistics.

Benoît Sagot. 2010. The Le*fff*, a freely available and large-coverage morphological and syntactic lexicon for french. In *Proceedings of LREC'10*, Valetta, Malta.

Djamé Seddah, Marie Candito, and Benoit Crabbé. 2009. Cross parser evaluation and tagset variation: A French Treebank study. In *Proceedings of the 11th Internation Conference on Parsing Technologies (IWPT'09)*, pages 150–161, Paris, France, October. Association for Computational Linguistics.

Djamé Seddah, Grzegorz Chrupała, Ozlem Cetinoglu, Josef van Genabith, and Marie Candito. 2010. Lemmatization and statistical lexicalized parsing of morphologically-rich languages. In *Proceedings of the NAACL/HLT Workshop on Statistical Parsing of Morphologically Rich Languages (SPMRL 2010)*, Los Angeles, CA.

Yannick Versley and Ines Rehbein. 2009. Scalable discriminative parsing for german. In *Proceedings of the 11th International Conference on Parsing Technologies (IWPT'09)*, pages 134–137, Paris, France, October. Association for Computational Linguistics.