Improving generative statistical parsing with semi-supervised word clustering

Marie Candito, Benoît Crabbé

Improving generative statistical parsing with semi-supervised word clustering

Marie Candito and Benoît Crabbé
Université Paris 7/INRIA (Alpage), 30 rue du Château des Rentiers, 75013 Paris

Abstract

We present a semi-supervised method to improve statistical parsing performance. We focus on the well-known problem of lexical data sparseness and present experiments of word clustering prior to parsing. We use a combination of lexicon-aided morphological clustering that preserves tagging ambiguity, and unsupervised word clustering, trained on a large unannotated corpus. We apply these clusterings to the French Treebank, and we train a parser with the PCFG-LA unlexicalized algorithm of (Petrov et al., 2006). We find a gain in French parsing performance: from a baseline of F_1=86.76% to F_1=87.37% using morphological clustering, and up to F_1=88.29% using further unsupervised clustering. This is the best known score for French probabilistic parsing. These preliminary results are encouraging for statistically parsing morphologically rich languages, and languages with small amount of annotated data.

1 Introduction

Lexical information is known crucial in natural language parsing. For probabilistic parsing, one main drawback of the plain PCFG approach is to lack sensitivity to the lexicon. The symbols accessible to context-free rules are part-of-speech tags, which encode generalizations that are too coarse for many parsing decisions (for instance subcategorization information is generally absent from tagsets). The lexicalized models first proposed by Collins reintroduced words at every depth of a parse tree, insuring that attachments receive probabilities that take lexical information into account. On the other hand, (Matsuzaki et al., 2005) have proposed probabilistic CFG learning with latent annotation (hereafter PCFG-LA), as a way to automate symbol splitting in unlexicalized probabilistic parsing (cf. adding latent annotations to a symbol is comparable to splitting this symbol). (Petrov et al., 2006) rendered the method usable in practice, with a tractable technique to retain only the beneficial splits.

We know that both lexicalized parsing algorithm and PCFG-LA algorithm suffer from lexical data sparseness. For lexicalized parsers, (Gildea, 2001) shows that bilexical dependencies parameters are almost useless in the probabilistic scoring of parser because they are too scarce.
For PCFG-LA, we have previously studied the lexicon impact on this so-called “unlexicalized” algorithm, for French parsing (Crabbé and Candito, 2008), (Candito et al., 2009). We have tested a totally unlexicalized parser, trained on a treebank where words are replaced by their POS tags. It obtains a parseval F_1=86.28 (note that it induces perfect tagging). We compared it to a parser trained with word+tag as terminal symbols (to simulate a perfect tagging), achieving F_1=87.79. This proves that lexical information is indeed used by the “unlexicalized” PCFG-LA algorithm: some lexical information percolates through parse trees via the latent annotations.

We have also reported a slight improvement (F_1=88.18) when word forms are clustered on a morphological basis, into lemma+tag clusters. So PCFG-LA uses lexical information, but it is too sparse, hence it benefits from word clustering. Yet the use of lemma+tag terminals supposes tagging prior to parsing. We propose here to apply rather a deterministic supervised morphological clustering that preserves tagging ambiguities, leaving it to the parser to disambiguate POS tags.

We also investigate the use of unsupervised word clustering, obtained from unannotated text. It has been proved useful for parsing by (Koo et al., 2008) and their work directly inspired ours. They have shown that parsing improves when cluster information is used as features in a discriminative training method that learns dependency parsers. We investigate in this paper the use of such clusters in a generative approach to probabilistic phrase-structure parsing, simply by replacing each token by its cluster.

We present in section 2 the treebank instantiation we use for our experiments, the morphological clustering in section 3, and the Brown algorithm for unsupervised clustering in section 4. Section 5 presents our experiments, results and discussion. Section 6 discusses related work. Section 7 concludes with some ideas for future work.

2 French Treebank

For our experiments, we use the French Treebank (hereafter FTB) (Abeillé et al., 2003), containing 12531 sentences of the newspaper *Le Monde*. We started with the treebank instantiation defined in (Crabbé and Candito, 2008), where the rich original annotation containing morphological and functional information is mapped to a plain phrase-structure treebank with a tagset of 28 POS tags.

In the original treebank, 17% of the tokens belong to a compound, and compounds range from very frozen multi word expressions like *y compris* (literally *there included*, meaning *including*) to syntactically regular entities like *loi agraire* (*land law*). In most of the experiments with the FTB, each compound is merged into a single token: `(P (CL y) (A compris))` is merged as `(P y_compris)`. But because our experiments aim at reducing lexical sparseness but also at augmenting lexical coverage using an unannotated corpus, we found it necessary to make the unannotated corpus tokenisation and the FTB tokenisation consistent. To set up a robust parser, we chose to avoid recognizing compounds that exhibit syntactically regular patterns. We create a new instance of the treebank (hereafter FTB-UC), where syntactically regular patterns are "undone" (Figure 1). This reduces the number of distinct compounds in the whole treebank from 6125 to 3053.

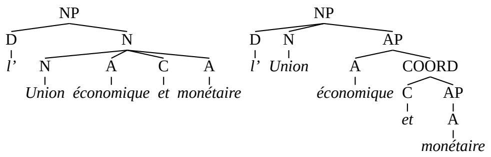

Figure 1: A NP with a compound (left) changed into a regular structure with simple words (right)

3 Morphological clustering

The aim of this step is to reduce lexical sparseness caused by inflection, without hurting parsability, and without committing ourselves as far as ambiguity is concerned. Hence, a morphological clustering using lemmas is not possible, since lemma assignment supposes POS disambiguation. Further, information such as mood on verbs is necessary to capture for instance that infinitive verbs have no overt subject, that participial clauses are sentence modifiers, etc... This is encoded in the FTB with different projections for finite verbs (projecting sentences) versus non finite verbs (projecting VPpart or VPinf).

We had the intuition that the other inflection marks in French (gender and number for determiners, adjectives, pronouns and nouns, tense and person for verbs) are not crucial to infer the correct phrase-structure projected by a given word[1].

So to achieve morphological clustering, we designed a process of *desinflection*, namely of removing some inflection marks. It makes use of the Lefff, a freely available rich morphological and syntactic French lexicon (Sagot et al., 2006), containing around 116000 lemmas (simple and compounds) and 535000 inflected forms. The desinflection is as follows: for a token t to *desinflect*, if it is known in the lexicon, for all the inflected lexical entries le of t, try to get corresponding singular entries. If for all the le, corresponding singular entries exist and all have the same form, then replace t by the corresponding singular. 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*.

Then the same process applies to map feminine forms to corresponding masculine forms. This allows 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 ambiguity, *entrée* is not replaced by *entré*. Finite verb forms, when unambiguous with other POS, are mapped to second person plural present indicative corresponding forms. This choice was made in order to avoid creating ambiguity: the second person plural forms end with a very typical *-ez* suffix, and the resulting form is very unlikely ambiguous. For the first

[1]For instance, French oral comprehension does not seem to need plural marks very much, since a majority of French singular forms have their corresponding plural form pronounced in the same way.

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

token of a sentence, if unknown in the lexicon, the algorithm tries to desinflect the low case corresponding form.

This desinflection reduces the number of distinct tokens in the FTB-UC from 27143 to 20268.

4 Unsupervised word clustering

We chose to 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) or 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... This results in a hard clustering into C clusters. The process can be continued to further merge pairs of clusters among the C clusters, ending with a unique cluster for the whole vocabulary. This can be traced to obtain a binary tree representing the merges of the C clusters. A cluster can be identified by its path within this binary tree. Hence, clusters can be used at various levels of granularity.

5 Experiments and discussion

For the Brown clustering algorithm, we used Percy Liang's code[3], run on the *L'Est Républicain* corpus, a 125 million word journalistic corpus, freely available at CNRTL[4]. The corpus was tokenised[5], segmented into sentences and desinflected using the process described in section 3. We ran the clustering into 1000 clusters for the desinflected forms appearing at least 20 times.

We tested the use of word clusters for parsing with the Berkeley algorithm (Petrov et al., 2006). Clustering words in this case has a double advantage. First, it augments the known vocabulary, which is made of all the forms of all the clusters appearing in the treebank. Second, it reduces sparseness for the latent annotations learning on the lexical rules of the PCFG-LA grammar.

We used Petrov's code, adapted to French by (Crabbé and Candito, 2008), for the suffixes used to classify unknown words, and we used the same training(80%)/dev(10%)/test(10%) partition. We used the FTB-UC treebank to train a baseline parser, and three other parsers by changing the terminal symbols used in training data:

desinflected forms: as described in section 3

clusters + cap: each desinflected form is replaced by its cluster bit string. If the desinflected form has no corresponding cluster (it did not appear 20 times in the unannotated corpus), a special cluster UNKC is used. Further, a _C suffix is added if the form starts with a capital.

clusters + cap + suffixes: same as before, except that 9 additional features are used as suffixes to the cluster: if form is all digits, ends with *ant*, or *r*, or *ez* (cf. this is how end desinflected forms of unambiguous finite verbs), ...

We give in table 1 parsing performance in terms of labeled precision/recall/Fscore, and also the more neutral unlabeled attachment score (UAS)[6].

The desinflection process does help: benefits from reducing data sparseness exceed the loss of agreement markers. Yet tagging decreases a little, and this directly impacts the dependency score, because the dependency extraction uses head propagation rules that are sensitive to tagging. In the same way, the use of bare clusters increases labeled recall/precision, but the tagging accuracy decreases, and thus the UAS. This can be due to the coarseness of the clustering method, which sometimes groups words that have different POS (for instance among a cluster of infinite verbs, one may find a present participle). The quality of the clusters is more crucial in our case than when clusters are features, whose informativity is discriminatively learnt. This observation led us to append a restricted set of suffixes to the clusters, which gives us the best results for now.

6 Related work

We already mentioned that we were inspired by the success of (Koo et al., 2008) in using word clusters as features for the discriminative learning of dependency parsers. Another approach to augment the known vocabulary for a generative prob-

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

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

[5] The 200 most frequent compounds of the FTB-UC were systematically recognized as one token.

[6] In all metrics punctuation tokens are ignored and all results are for sentences of less than 40 words. Note that we used the FTB-UC treebank. There are mors tokens in sentences than in the FTB with all compounds merged, and baseline F_1 scores are a little higher (86.79 versus 86.41).

terminal symbols	LP	LR	F_1	UAS	Vocab. size	Tagging Acc.
inflected forms (baseline)	86.94	86.65	86.79	91.00	27143	96.90
desinflected forms	87.42	87.32	87.37	91.14	20268	96.81
clusters + cap	88.08	87.50	87.79	91.12	1201	96.37
clusters + cap + suffixes	88.43	88.14	**88.29**	**91.68**	1987	**97.04**

Table 1: Parsing performance when training and parsing use clustered terminal symbols

abilistic 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 proved useful to better parse Hebrew.

7 Conclusion and future work

We have tested the very simple method of replacing inflected forms by clusters of forms in a generative probabilistic parser. This crude technique has surprisingly good results and offers a very cheap and simple way to augment the vocabulary seen at training time. It seems interesting to try the technique on other generative approaches such as lexicalized probabilistic parsing.

We plan to optimize the exact shape of terminal symbols to use. Bare unsupervised clusters are unsatisfactory, and we have seen that adding simple suffixes to the clusters improved performance. Learning such suffixes is a path to explore. Also, the hierarchical organization of the clusters could be used, in the generative approach adopted here, by modulating the granularity of the clusters depending on their frequency in the treebank.

We also need to check to what extent the desinflection step helps for taking advantage of the very local information captured by the Brown clustering.Finally, we could try using other kinds of clustering, such as the approach of (Lin, 1998), which captures similarity between syntactic dependencies beared by nouns and verbs.

8 Acknowledgements

The authors truly thank Percy Liang and Slav Petrov for providing their code for respectively Brown clustering and PCFG-LA. This work was supported by the French National Research Agency (SEQUOIA project ANR-08-EMER-013).

References

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

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, Benoit Crabbé, and Djamé Seddah. 2009. On statistical parsing of french with supervised and semi-supervised strategies. In *EACL 2009 Workshop Grammatical inference for Computational Linguistics*, Athens, Greece.

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.

Daniel Gildea. 2001. Corpus variation and parser performance. In *Proc. of EMNLP'01*, pages 167–202, Pittsburgh, USA.

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.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proc. of ACL-08*, Columbus, USA.

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

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of ACL-98*, pages 768–774, Montreal, Canada.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic cfg with latent annotations. In *Proc. of ACL-05*, pages 75–82, Ann Arbor, USA.

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

Benoît Sagot, Lionel Clément, Éric Villemonte de La Clergerie, and Pierre Boullier. 2006. The Lefff 2 syntactic lexicon for french: Architecture, acquisition, use. In *Proc. of LREC'06*, Genova, Italy.