Coupling an Annotated Corpus and a Morphosyntactic Lexicon for State-of-the-Art POS Tagging with Less Human Effort

Pascal Denis and Benoît Sagot

Equipe-project ALPAGE
INRIA and Université Paris 7
30, rue Château des Rentiers
75013 Paris, France
{pascal.denis,benoit.sagot}@inria.fr

Abstract. This paper investigates how to best couple hand-annotated data with information extracted from an external lexical resource to improve POS tagging performance. Focusing on French tagging, we introduce a maximum entropy conditional sequence tagging system that is enriched with information extracted from a morphological resource. This system gives a 97.7% accuracy on the French Treebank, an error reduction of 23% (28% on unknown words) over the same tagger without lexical information. We also conduct experiments on datasets and lexicons of varying sizes in order to assess the best trade-off between annotating data vs. developing a lexicon. We find that the use of a lexicon improves the quality of the tagger at any stage of development of either resource, and that for fixed performance levels the availability of the full lexicon consistently reduces the need for supervised data by at least one half.

Keywords: Part-of-speech tagging, maximum entropy models, morphosyntactic lexicon, French, language resource development

1 Introduction

Over recent years, numerous systems for automatic part-of-speech (POS) tagging have been proposed for a large variety of languages. Among the best performing systems are those based on supervised machine learning techniques (see (Manning and Schütze, 1999) for an overview). For some languages like English and other European languages, these systems have reached performance that comes close to human levels. Interestingly, the majority of these systems have been built without resorting to any external lexical information sources; they instead rely on a dictionary that is based on the training corpus (see however (Hajič, 2000)). This raises the question of whether we can still improve tagging performance by exploiting this type of resource. Arguably, a potential advantage of using an external dictionary is in a better handling of unknown words (i.e., words that are not present in the training corpus, but that may be present in the external dictionary). A subsequent question is how to best integrate the information from a lexical resource into a probabilistic POS tagger. In this paper, we consider two distinct scenarios: (i) using the external dictionary as *constraints* that restrict the set of possible tags that the tagger can choose from, and (ii) incorporating the dictionary tags as *features* in a probabilistic POS tagging model. Another interesting question is that of the relative impact of training corpora and of lexicons of various sizes. This issue is crucial to the development of POS taggers for resource-scarce languages for which it is important to determine the best trade-off between annotating data and constructing dictionaries.

This paper addresses these questions through various tagging experiments carried out on French, based on our new tagging system called MElt (Maximum-Entropy Lexicon-enriched Tagger). An obvious motivation for working on this language is the availability of a training corpus

(namely, the French Treebank (Abeillé *et al.*, 2003)) and a large-scale lexical resource (namely, Le*fff* (Sagot *et al.*, 2006)). Additional motivation comes from the fact that there has been comparatively little work in probabilistic POS tagging in this language. An important side contribution of our paper is the development of a state-of-the-art, freely distributed POS tagger for French.[1] Specifically, we here adopt a maximum entropy (MaxEnt) sequential classification approach. MaxEnt models remain among the best performing tagging systems for English and they are particularly easy to build and fast to train.

This paper is organized as follows. Section 2 describes the datasets and the lexical resources that were used. Section 3 presents a baseline MaxEnt tagger for French that is inspired by previous work, in particular (Ratnaparkhi, 1996) and (Toutanova and Manning, 2000), that already outperforms TreeTagger (Schmid, 1994) retrained on the same data. In Section 4, we show that the performance of our MaxEnt tagger can be further improved by incorporating features extracted from a large-scale lexicon, reaching a 97.7% accuracy, which compares favorably with the best results obtained for English with a similar tagset. Finally, Section 5 evaluates the relative on accuracy impact of the training data and the lexicon during tagger development by varying their respective sizes.

2 Resources and tagset

2.1 Corpus

The morphosyntactically annotated corpus we used is a variant of the French TreeBank or FTB, (Abeillé *et al.*, 2003). It differs from the original FTB in so far that all compounds that do not correspond to a syntactically regular sequence of categories have been merged into unique tokens and assigned a category corresponding to their spanning node; other compounds have been left as sequences of several tokens (Candito, p.c.). The resulting corpus has $350,931$ tokens in $12,351$ sentences.

In the original FTB, words are split into 13 main categories, themselves divided into 34 subcategories. The version of the treebank we used was obtained by converting subcategories into a tagset consisting of 28 tags, with a granularity that is intermediate between categories and subcategories. Basically, these tags enhance main categories with information on the mood of verbs and a few other lexical features. This expanded tagset has been shown to give the best statistical parsing results for French (Crabbé and Candito, 2008).[2] A sample tagged sentence from the FTB is given in Figure 1.

Cette/DET mesure/NC ,/PONCT qui/PROREL pourrait/V être/VINF appliquée/VPP dans/P les/DET prochaines/ADJ semaines/NC ,/PONCT permettrait/V d'/P économiser/VINF quelque/DET 4/ADJ milliards/NC de/P francs/NC ./PONCT

Figure 1: Sample data from FTB in Brown format

As in (Candito *et al.*, 2009), the FTB is divided into 3 sections : training (80%), development (10%) and test (10%). The dataset sizes are presented in Table 1 together with the number of unknown words.

2.2 Lexicon

One of the goals of this work is to study the impact of using an external dictionary for training a tagger, in addition to the training corpus itself. We used the morphosyntactic information included in the large-coverage morphological and syntactic lexicon Le*fff* (Sagot *et al.*, 2006).[3]

[1] The MElt tagger is freely available from `http://gforge.inria.fr/projects/lingwb/`.

[2] This tagset is known as TREEBANK+ in (Crabbé and Candito, 2008), and since then as CC (Candito *et al.*, 2009).

[3] The Le*fff* is freely distributed under the LGPL-LR license at `http://alexina.gforge.inria.fr/`

Table 1: Data sets

Data Set	# of sent.	# of tokens	# of unk. tokens
FTB-TRAIN	$9,881$	$278,083$	
FTB-DEV	$1,235$	$36,508$	$1,892$ (5.2%)
FTB-TEST	$1,235$	$36,340$	$1,774$ (4.9%)

Although Le*fff* contains both morphological and syntactic information for each entry (including sub-categorization frames, in particular for verbs), we extracted only the morphosyntactic information. We converted categories and morphological tags into the same tagset used in the training corpus, hence building a large-coverage morphosyntactic lexicon containing $507,362$ distinct entries of the form *(form, tag, lemma)*, corresponding to $502,223$ distinct entries of the form *(form, tag)*. If grouping all verbal tags into a single "category" while considering all tags as "categories", these entries correspond to $117,397$ *(lemma, category)* pairs (the relevance of these pairs will appear in Section 5).

3 Baseline MaxEnt tagger

This section presents our baseline MaxEnt-based French POS tagger, MElt$^0_{\text{fr}}$. This tagger is largely inspired by (Ratnaparkhi, 1996) and (Toutanova and Manning, 2000), both in terms of the model and the features being used. To date, MaxEnt conditional sequence taggers are still among the best performing taggers developed for English.[4] An important appeal of MaxEnt models is that they allow for the combination of very diverse, potentially overlapping features without assuming independence between the predictors. These models have also the advantage of being very fast to train.[5]

3.1 Description of the task

Given a tagset T and a string of words w_1^n, we define the task of tagging as the process of assigning the maximum likelihood tag sequence $\hat{t}_1^n \in T^n$ to w_1^n. Following (Ratnaparkhi, 1996), we can approximate the conditional probability $P(t_1^n|w_1^n)$ so that:

$$\hat{t}_1^n = \arg\max_{t_1^n \in T^n} P(t_1^n|w_1^n) \approx \arg\max_{t_1^n \in T^n} \prod_{i=1}^{n} P(t_i|h_i) \quad (1)$$

where t_i is the tag for word w_i, and h_i is the "history" (or context) for (w_i, t_i), which comprises the preceding tags t_i^{i-1} and the word sequence w_i^n.

3.2 Model and features

In a MaxEnt model, the parameters of an exponential model of the following form are estimated:

$$P(t_i|h_i) = \frac{1}{Z(h)} \cdot \exp\left(\sum_{i=j}^{m} \lambda_j f_j(h_i, t_i)\right) \quad (2)$$

f_1^m are feature functions defined over tag t_i and history h_i (with $f(h_i, t_i) \in \{0, 1\}$), λ_1^m are the parameters associated with f_1^m, and $Z(h)$ is a normalization term over the different tags. In this type of model, the choice of the parameters is subject to constraints that force the model expectations of the features to be equal to their empirical expectations over the training data (Berger *et al.*,

[4] (Ratnaparkhi, 1996) and (Toutanova and Manning, 2000) report accuracy scores of 96.43 and 96.86 on section 23-24 of the Penn Treebank, respectively.

[5] Arguably better suited for sequential problems, Conditional Random Fields (CRF) (Lafferty *et al.*, 2001) are considerably slower to train.

1996). In our experiments, the parameters were estimated using the Limited Memory Variable Metric Algorithm (Malouf, 2002) implemented in the Megam package.[6]

The feature templates we used for designing our French tagging model is a superset of the features used by (Ratnaparkhi, 1996) and (Toutanova and Manning, 2000) for English (these were largely language independent). These features fall into two main categories. A first set of features try to capture the *lexical form* of the word being tagged: these include the actual word string for the current word w_i, prefixes and suffixes (of character length 4 and less), as well as binary features testing whether w_i contains special characters like numbers, hyphens, and uppercase letters. A second set of features directly model the *context* of the current word and tag: these include the previous tag, the concatenation of the two previous tags, as well as the surrounding word forms in a window of 2 tokens.

The detailed list of feature templates we used in this baseline tagger is shown in Table 2.[7]

Table 2: Baseline model features

Lexical features	
$w_i = X$	& $t_i = T$
Prefix of $w_i = P, \|P\| < 5$	& $t_i = T$
Suffix of $w_i = S, \|S\| < 5$	& $t_i = T$
w_i contains number	& $t_i = T$
w_i contains hyphen	& $t_i = T$
w_i contains uppercase character	& $t_i = T$
w_i contains only uppercase characters	& $t_i = T$
w_i contains up. char. and doesn't start sentence	& $t_i = T$
Contextual features	
$t_{i-1} = X$	& $t_i = T$
$t_{i-2}t_{i-1} = XY$	& $t_i = T$
$w_{i+j} = X, j \in \{-2, -1, 1, 2\}$	& $t_i = T$

An important difference with (Ratnaparkhi, 1996) in terms of feature design is that we did not restrict the application of the prefix/suffix features to words that are rare in the training data. In our model, these features always get triggered, even for infrequent words. We found that the permanent inclusion of these features led to better performance during development, which can probably be explained by the fact that these features get better statistics and are extremely useful for unknown words. These features are also probably more discriminative in French than in English, since it is morphologically richer. Another difference to previous work regards smoothing. (Ratnaparkhi, 1996) and (Toutanova and Manning, 2000) use feature count cutoff of 10 to avoid unreliable statistics for rare features. We did not use cutoffs but instead use a regularization Gaussian prior on the weights[8], which is arguably a more principled smoothing technique.[9]

3.3 Testing and Performance

The test procedure relies on a *beam search* to find the most probable tag sequence for a given sentence. That is, each sentence is decoded from left to right and we maintain for each word w_i

[6] Available from `http://www.cs.utah.edu/~hal/megam/`.

[7] Recall that features in MaxEnt are functions ranging on both contexts and classes. A concrete example of one of our feature is given below:

$$f_{100}(h,t) = \begin{cases} 1 & \text{if } w_i = \text{"le" \& } t = \texttt{DET} \\ 0 & \text{otherwise} \end{cases}$$

[8] Specifically, we used a prior with precision (i.e., inverse variance) of 1 (which is the default in Megam); other values were tested during development but did not yield improvements.

[9] Informally, the effect of this kind of regularization is to penalize artificially large weights by forcing the weights to be distributed according to a Gaussian distribution with mean zero.

the n highest probability tag sequence candidates up to w_i. For our experiments, we used a beam size of 3.[10] In addition, the test procedure utilizes a *tag dictionary* which lists for a given word the tags associated with this word in the training data. This drastically restricts the allowable labels that the tagger can choose from for a given word, in principle leading to fewer tagging errors and reduced tagging time.

The maximum entropy tagger described above, MElt$^0_{\text{fr}}$, was compared against two other baseline taggers, namely: UNIGRAM and TreeTagger. UNIGRAM works as follows: for a word seen in the training corpus, this tagger uses the most frequent tag associated with this word in the corpus; for unknown words, it uses the most frequent tag in the corpus (in this case, `NC`). TreeTagger is a statistical, decision tree-based POS tagger (Schmid, 1994).[11] The version used for this comparison was retrained on the FTB training corpus. The performance results of the three taggers are given in Table 3; scores are reported in terms of accuracy over both the entire test set and the words that were not seen during training.

Table 3: Baseline tagger performance

Tagger	Overall Acc.	Unk. Word Acc.
UNIGRAM	91.90	24.50
TreeTagger	96.12	75.77
MElt$^0_{\text{fr}}$	**97.00**	**86.10**

As shown in Table 3, MElt$^0_{\text{fr}}$ achieves accuracy scores of 97% overall and 86.1% on unknown words.[12] Our baseline tagger significantly outperforms the retrained version of TreeTagger, with an improvement of over 10% on unknown words.[13] There are several possible explanations for such a discrepancy in handling unknown words. The first one is that MaxEnt parameter estimation is less prone to data fragmentation for sparse features than Decision Tree parameter estimation due to the fact that it does not partition the training sample. A second related explanation is that TreeTagger simply misses some of the generalizations regarding lexical features due to the fact that it only includes suffixes and this only for unknown words.

4 Lexicon-enriched MaxEnt tagger

For trying to further improve MElt$^0_{\text{fr}}$, we investigate in this Section the impact of coupling it with an external lexical resource, and compare two ways of integrating this new information: as contraints vs. as features.

4.1 Integrating lexical information in the tagger

The most natural way to make use of the extra knowledge supplied by a lexicon is to represent it as "filtering" constraints: that is, the lexicon is used as an additional tag dictionary guiding the POS tagger, in addition to the lexicon extracted from the training corpus. Under this scenario, the tagger is forced for a given word w to assign one of the tags associated with w in the full tag dictionary: the set of allowed tags for w is the union of the sets of its tags in the corpus and in Le*fff*. This approach is similar to that of (Hajič, 2000), who applied it to highly inflected languages, and in particular to Czech.

In a learning-based tagging approach, there is another possibility to accommodate the extra information provided by Le*fff*: we can directly incorporate the tags associated by Le*fff* to each

[10] We tried larger values (i.e., 5, 10, 15, 20) during development, but none of these led to significant improvements.

[11] Available at `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`.

[12] The accuracy results of MElt$^0_{\text{fr}}$ on FTB-DEV are: 96.7 overall and 86.2 on unknown words.

[13] Chi-square statistical significance tests were applied to changes in accuracy, with p set to 0.01 unless otherwise stated.

word in the form of features. Specifically, for each word, we posit a new lexical feature for each of its possible tags according to the Le*fff*, as well as a feature that represents the disjunction of all Le*fff* tags (provided there is more than one). Similarly, we can also use the Le*fff* to provide additional contextual features: that is, we can include Le*fff* tags for all the words in a window of 2 tokens. Table 4 summarizes these new feature templates.

Table 4: Lexicon-based features

Lexical features	
Le*fff* tag for $w_i = X$	& $t_i = T$
Le*fff* tags for $w_i = X_0 \dots X_n$	& $t_i = T$
Contextual features	
Le*fff* tag for $w_{i+j} = X, j \in \{-2, -1, 1, 2\}$	& $t_i = T$
Le*fff* tags for $w_{i+j} = X_0 \dots X_n, j \in \{-2, -1, 1, 2\}$	& $t_i = T$

Integrating the lexical information in this way has a number of potential advantages. First, features are by definition more robust to noise (in this case, to potential errors in the lexicon or simply mismatches between the corpus annotations and the lexicon categories). Furthermore, some the above features directly model the context, while the filtering constraints are entirely non contextual.

4.2 Comparative evaluation

We compared the performance of the Le*fff*-constraints based tagger $\text{MElt}^c_{\text{fr}}$ and Le*fff*-features based tagger $\text{MElt}^c_{\text{fr}}$ to other lexicon-enriched taggers. The first of these taggers, $\text{UNIGRAM}_{\text{Le}fff}$, like UNIGRAM, is a unigram model based on the training corpus, but it uses Le*fff* for labeling unknown words: among the possible Le*fff* tag for a word, this model chooses the tag that is most frequent in the training corpus (all words taken into account). Words that are unknown to both the corpus and Le*fff* are assigned NC. The second tagger, $\text{TreeTagger}_{\text{Le}fff}$ is a retrained version of TreeTagger to which we provide Le*fff* as an external dictionary. Finally, we also compare our tagger to F-BKY, an instantiation of the Berkeley parser adapted for French by (Crabbé and Candito, 2008) and used as a POS tagger. The performance results for these taggers are given in Table 5.

Table 5: Lexicon-based taggers performance

Tagger	Overall Acc.	Unk. Word Acc.
$\text{UNIGRAM}_{\text{Le}fff}$	93.40	55.00
$\text{TreeTagger}_{\text{Le}fff}$	96.55	82.14
F-BKY	97.30	82.90
MElt	97.00	86.10
$\text{MElt}^c_{\text{fr}}$	97.10	89.00
$\text{MElt}^f_{\text{fr}}$	**97.70**	**90.01**

The best tagger is $\text{MElt}^f_{\text{fr}}$, with accuracy scores of 97.7% overall and 90.1% for unknown words. This represents significant improvements of .7% and 3.9% over MElt, respectively. These scores put $\text{MElt}^f_{\text{fr}}$ above all the other taggers we have tested, including the parser-based F-BKY, by a significant margin. To our knowledge, these scores are the best scores reported for French POS tagging.[14] By contrast, $\text{MElt}^c_{\text{fr}}$ achieves a rather limited (and statistically insignificant) performance gain of .1% overall but a 2.9% improvement on unknown words.

[14] The accuracy results of $\text{MElt}^f_{\text{fr}}$ on FTB-DEV are: 97.3 overall and 90.01 on unknown words.

Our explanation for these improvements is that the Le*fff*-based features reduce data sparseness and provide useful information on the right context: first, fewer errors on unknown words (a direct result of the use of a morphosyntactic lexicon) necessarily leads to fewer erroneous contexts for other words, and therefore to better tagging; second, the possible categories of tokens that are on the right of the current tokens are valuable pieces of information, and they are available only from the lexicon. The lower result of $\text{MElt}^{c}_{\text{fr}}$ can probably be explained by two differences: it does not benefit from this additional information about the right context, and it uses Le*fff* information as hard constraints, not as (soft) features.

4.3 Error analysis

In order to understand whether the 97.7% accuracy of $\text{MElt}^{f}_{\text{fr}}$ could still be improved, we decided to examine manually its first 200 errors on FTB-DEV, and classify them according to an adequate typology of errors. The resulting typology and the corresponding figures are given in Table 6.

Table 6: Manual error analysis of the 200 first errors of $\text{MElt}^{f}_{\text{fr}}$ on the development corpus

Error type		Frequency
Standard errors	Adjective vs. past participle	5.5%
	Errors on *de*, *du*, *des*	4.0%
	Other errors	34%
Errors on numbers		15.5%
Errors related to named entities		27.5%
$\text{MElt}^{f}_{\text{fr}}$'s result seem correct	Error in FTB-DEV	8.5%
	Unclear cases (both tags seem valid)	4.5%
	Truncated text in FTB-DEV	0.5%

These results show that the 97.7% score can still be improved. Indeed, standard named entity recognition techniques could help solve most errors related to named entities, i.e., more than one out of four errors. Moreover, simple regular patterns could allow for replacing automatically all numbers by one or several placeholder(s) both in the training and evaluation data. Indeed, preserving numbers as such inevitably leads to a sparse data problem, which prevents the training algorithm from modeling the complex task of tagging numbers — they can be determiners, nouns, adjectives or pronouns. Appropriate placeholders should significantly help the training algorithm and improve the results. Finally, no less than 13.5% of $\text{MElt}^{f}_{\text{fr}}$'s apparent errors are in fact related to FTB-DEV's annotation, because of errors (9%) or unclear situations, for which both the gold tag and $\text{MElt}^{f}_{\text{fr}}$'s tag seem valid.

Given these facts, we consider feasible to improve $\text{MElt}^{f}_{\text{fr}}$ from 97.7% to 98.5% in the future.

5 Varying training corpus and lexicon sizes

5.1 Motivations and experimental setup

The results achieved in the previous section have been made possible by the (relatively) large size of the corpus and the broad coverage of the lexicon. However, such resources are not always available for a given language, in particular for so-called under-resourced languages. Moreover, the significant improvement observed by using Le*fff* shows that the information contained in a morphosyntactic lexicon is worth using. The question arises to know if this lexical information is able to compensate for the lack of a large training corpus. Symmetrically, it is unclear how various lexicon sizes impact the quality of the results.

Therefore, we performed a series of experiments by training $\text{MElt}^{f}_{\text{fr}}$ on various subcorpora and sublexicons. Extracting subcorpora is simple: the first s sentences of a training corpus constitute

a reasonable corpus of size s. However, extracting sublexicons is less trivial. We decided to extract increasingly large sublexicons in a way that approximately simulates the development of a morphosyntactic lexicon. To achieve this goal, we used the MElt$^{f}_{\text{fr}}$ tagger described in the previous section to tag a large raw corpus.[15] We then lemmatized the corpus by assigning to each token the list of all of its possible lemmas that exhibit a category consistent with the annotation. Finally, we ranked all resulting *(lemma, category)* pairs w.r.t. frequency in the corpus. Extracting a sublexicon of size n then consists in extracting all *(form,tag,lemma)* entries whose corresponding *(lemma, category)* pair is among the l best ranked ones.

We reproduced the same experiments as those described in Section 4, but training MElt$^{f}_{\text{fr}}$ on various subcorpora and various sublexicons. We used 9 different lexicon sizes and 8 different corpus sizes, summed up in Table 7. For each resulting tagger, we measured the overall accuracy and the accuracy on unknown words.

Table 7: Varying training corpus and lexicon sizes: experimental setups

Lexicon size (lemmas)	0, 500, $1,000$, $2,000$, $5,000$, $10,000$, $20,000$, $50,000$, $110,000$
Corpus size (sentences)	50, 100, 200, 500, $1,000$, $2,000$, $5,000$, $9,881$

5.2 Results and discussion

Before comparing the respective relevance of lexicon and corpus manual development for optimizing the tagger's performance, we need to be able to quantitatively compare their development costs, i.e., times.

In (Marcus *et al.*, 1993), the authors report a POS annotation speed that "exceeds $3,000$ words per hour" during the development of the Penn TreeBank. This speed is reached after a 1 month period (with 15 annotation hours per week, i.e., approximately 60 hours) during which the POS tagger used for pre-annotation was still improving. The authors also report on a manual tagging experiment (without automatic pre-annotation); they observed an annotation speed that is around $1,300$ words per hour. Therefore, it is probably safe to assume that, on average, the creation of a manually validated training corpus starts at a speed that is around 1,000 words (30 sentences) per hour, and increases up to $3,000$ words (100 sentences) per hour once the corpus has reached, say, $5,000$ sentences.

For lexicon development, techniques such as those described in (Sagot, 2005) allow for a fast validation of automatically proposed hypothetical lemmas. Manual intervention is then limited to validation steps that take around 2 to 3 seconds per lemma, i.e., about $1,500$ lemmas per hour.

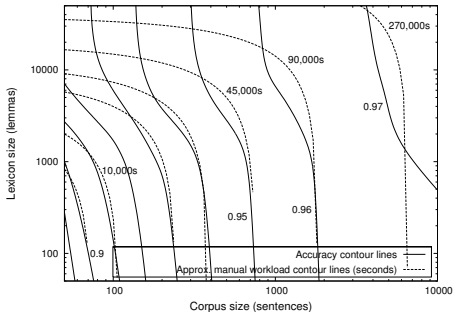
Figure 2 compares contour lines[16] for two functions of corpus and lexicon sizes: tagger accuracy and development time.[17] These graphs show different things:

- during the first steps of development (less than 3 hours of manual work), the distribution of the manual work between lexicon and corpus development has no significant impact on overall tagging accuracy, but accuracy on unknown words is better when focusing more or equally on the lexicon than on the corpus;

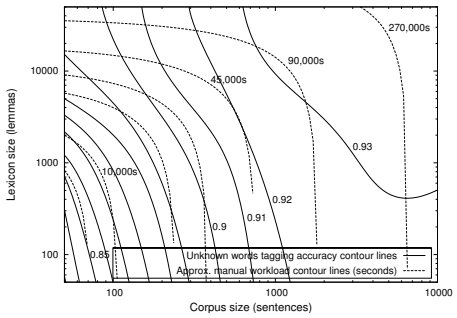
[15] We used a corpus of 20 million words extracted from the *L'Est Républicain* journalistic corpus, freely available at the web site of the CNRTL (`http://www.cnrtl.fr/corpus/estrepublicain/`).

[16] As computed by the `bspline` mode of `gnuplot`'s contour lines generation algorithm.

[17] The development times per sentence and per lexical entry mentioned in the previous paragraphs lead to the following formula for the total development time $t(s,l)$ (expressed in seconds), in which s is the number of sentences, l the number of lexical entries: $t(s,l) = 36s + 8400 \cdot log(s/100 + 1) + 2.4 \cdot l$.



(a)



(b)

Figure 2: Contour lines for two functions of corpus and lexicon sizes: tagger accuracy and development time. In (a), the tagger accuracy is measured overall, whereas in (b) it is restricted to unknown words

- in later stages of development, the optimal approach is to develop *both* the lexicon and the corpus, and this is true for both overall and unknown words tagging accuracy; however, it is by far better to concentrate only on corpus than only on lexicon development;
- using a morphological lexicon drastically improves the tagging accuracy on unknown words, whatever the development stage;
- for fixed performance levels, the availability of the full lexicon consistently reduces for training data by at least one half (and up to two thirds).

These results demonstrate the relevance of developing and using a morphosyntactic lexicon for improving tagging accuracy both in the early stages of development and for long-term optimization.

6 Conclusions and perspectives

We have introduced a new MaxEnt-based tagger, MElt, that we trained on the FTB for building a tagger for French. We show that this baseline can be significantly improved by coupling it with the French morphosyntactic lexicon Le*fff*. The resulting tagger, $\text{MElt}^{f}_{\text{fr}}$, reaches a 97.7% accuracy that are, to our knowledge, the best figures reported for French tagging, including parsing-based taggers. More precisely, the addition of lexicon-based features yield error reductions of 23.3% overall and of 27.5% for unknown words (corresponding to accuracy improvements of $.7\%$ and 3.9%, respectively) compared to the baseline tagger.

We also showed that the use of a lexicon improves the quality of the tagger at any stage of lexicon and training corpus development. Moreover, we approximately estimated development times for both resources, and show that the best way to optimize human work for tagger development is to work on the development of both an annotated corpus and a morphosyntactic lexicon.

In future work, we plan on trying and demonstrating this result in practice, by developing such resources and the corresponding $\text{MElt}^{f}_{\text{fr}}$ tagger for an under-resourced language. We also intend to study the influence of the tagset, in particular by training taggers based on larger tagsets. This work should try and understand how to benefit as much as possible from the internal structure of tags in such tagsets (gender, number, etc.).

References

Abeillé, A., L. Clément, and F. Toussenel. 2003. Building a treebank for French. In A. Abeillé, editor, *Treebanks*. Kluwer, Dordrecht.

Berger, A., S. D. Pietra, and V. D. Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Candito, M., B. Crabbé, and D. Seddah. 2009. On statistical parsing of French with supervised and semi-supervised strategies. In *Proceedings of the EACL'09 workshop on Grammatical Inference for computational linguistics*, Athens, Greece.

Crabbé, B. and M. Candito. 2008. Expériences d'analyses syntaxique statistique du français. In *Proceedings of TALN'08*, Avignon, France.

Hajič, J. 2000. Morphological Tagging: Data vs. Dictionaries. In *Proceedings of ANLP'00*, pages 94–101, Seattle, WA, USA.

Lafferty, J. D., A. McCallum, and F. C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Malouf, R. 2002. A comparison of algorithms for maximum entropy parameter estimation. In *Proceedings of the Sixth Workshop on Natural Language Learning*, pages 49–55, Taipei, Taiwan.

Manning, C. D. and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Marcus, M. P., M. A. Marcinkiewicz, and B. Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of International Conference on Empirical Methods in Natural Language Processing*, pages 133–142.

Sagot, B. 2005. Automatic acquisition of a Slovak lexicon from a raw corpus. In *Lecture Notes in Artificial Intelligence 3658, Proceedings of TSD'05*, pages 156–163, Karlovy Vary, Czech Republic. Springer-Verlag.

Sagot, B., L. Clément, É. de La Clergerie, and P. Boullier. 2006. The Le*fff* 2 syntactic lexicon for French: architecture, acquisition, use. In *Proceedings of LREC'06*, Lisbon, Portugal.

Schmid, H. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Toutanova, K. and C. D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of International Conference on New Methods in Language Processing*, pages 63–70, Hong Kong.