Tagging Unknown Proper Names Using Decision Trees

Frédéric Béchet † and **Alexis Nasr** ‡ and **Franck Genet**†
† LIA Université d'Avignon (frederic.bechet@lia.univ-avignon.fr)
‡ LIM Université Aix-Marseille 2 (alexis.nasr@lim.univ-mrs.fr)

Abstract

This paper describes a supervised learning method to automatically select from a set of noun phrases, embedding proper names of different semantic classes, their most distinctive features. The result of the learning process is a decision tree which classifies an unknown proper name on the basis of its context of occurrence. This classifier is used to estimate the probability distribution of an out of vocabulary proper name over a tagset. This probability distribution is itself used to estimate the parameters of a stochastic part of speech tagger.

1 Introduction

The work described in this paper aims at enriching lexica with new proper names. In such lexica, every word w is assigned a count distribution over the different tags of the tagger tagset (the number of times w was labelled with tag t). This distribution is called the *count distribution* of the word. The produced lexica are used to estimate the parameters of a POS stochastic tagger.

We will concentrate on proper names in a newspaper corpus (*Le Monde 1987-1992*), although the techniques described can be used for any category of words. The decision to concentrate on proper names follows from the fact that although proper names represent only a moderate proportion of the words occurrences in such corpora (3.67%), the probability of an out of vocabulary (OOV) word being a proper name is high. (Béchet and Yvon, 2000) report experiments concerning OOV proper names on a very close corpus (*Le monde diplomatique 1987-1995*). They showed that 72% of OOV words with respect to a newspaper 265 K words lexicon are potentially proper names. Furthermore, the same experiments showed that 30% of the sentences contain at least one OOV word. Besides, proper names are important for many tasks as information retrieval or named entities extraction.

The technique described has two stages. During the first stage, a training corpus is used to grow decision trees of a special kind called Semantic Classification Trees (SCT). Such trees model the salient features of the contexts in which words of a given semantic class occur. In a second stage, SCTs are used to update the lexical entries of OOV words appearing in a test corpus, based on their different context of occurrence in the test corpus. The updated lexicon is then used to estimate the parameters of a POS tagger.

The paper is structured as follows. In section 2 the tagger and the tagset used are described, section 3 introduces Semantic Classification Trees, section 4 describes how an SCT is built and section 5 how it is used to estimate the parameters of a POS tagger. The performances of the method are given in section 5.1, section 6 briefly describes other approaches to deal with OOV words and section 7 concludes the paper with some future work.

2 The tagger and the tagset

The POS tagger we use (Spriet and El-bèze, 1995) is based on the standard trigram model(Charniak et al., 1993):

$$\mathcal{T}(w_{1,n}) = arg \max_{t_{1,n}} \prod_{i=1}^{n} P(t_i|t_{i-2}t_{i-1})P(w_i|t_i) \tag{1}$$

where $\mathcal{T}(w_{1,n})$ is a sequence of n POS tags corresponding to the sequence of words $w_{1,n}$. The second term of the product of equation 1 ($P(w_i|t_i)$) is estimated using the count distributions stored in the lexicon.

The tagger was trained on the newspaper *Le Monde* between years 87 and 91. It uses a $265K$

words lexicon. The tagset consists of 105 morphosyntactic tags which include the usual major wordclasses plus some semantic subclasses for proper names: first names (FIRST), family names (FAMILY), countries, (COUNTRY) towns (TOWN) and organisations (ORG). We will primarily be interested in this paper by this subset of 5 tags which constitutes the semantic part of the tagset. When an OOV word is processed by the tagger, it is tagged UNK, for *unknown*.

This tagger have been evaluated on a hand-coded corpus. Its performances are comparable to other state-of-the-art systems (about 95% of accuracy).

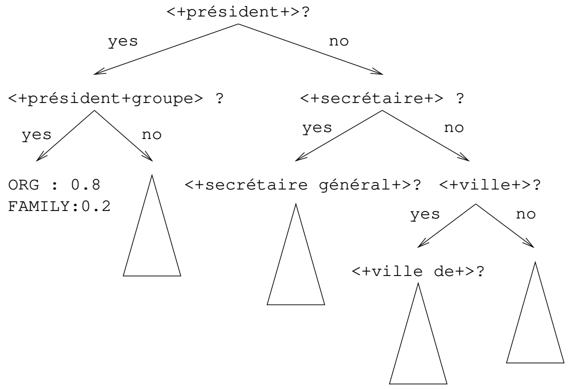

Figure 1: A semantic classification tree

3 Semantic Classification Trees

Semantic Classification Trees (SCT) have been introduced by (Kuhn and de Mori, 1996) as a means of classifying new strings from a corpus of tagged strings. We use it as a means of classifying noun phrases (NPs) containing unknown proper names from a corpus of labelled NPs. The NPs are labelled by the category of the embedded proper name. The label of an NP is therefore only related to the proper name included in it and not to the entity represented by the whole NP. For example, the NP : *the president of Uruguay* will be labelled with the tag COUNTRY and not PERSON.

Each node of the tree is associated with a regular expression (actually, only a limited form of regular expressions) involving lexical items, POS tags, gaps (non empty sequences of words or POS tags) and the two symbols $<$ and $>$ respectively indicating the beginning and the end of an NP. Each leaf of the tree is associated with a probability distribution over the 5 semantic tags. The category having the highest probability in this distribution is called the *top category* of the leaf. When an NP containing an OOV proper name matches the regular expression of a leaf, the associated distribution gives an estimation of the *lexical distribution* (probability distribution over the semantic tagset) of the unknown proper name.

An SCT has been represented in figure 1. Each node is labelled with its corresponding regular expression. The leftmost leaf, for example, corresponds to the regular expression <+président+groupe+> where the + signs denote gaps. Such a regular expression matches NPs as *le président du groupe X* (*the president of the X group*). The probability distribution of the unknown word, marked *X*, in this context, is given in the leaf.

4 Growing the decision tree

In order to build a decision tree, one needs an sample corpus, a set of questions, a split criteria and a stop condition, each of which is described below.

4.1 The sample corpus

The sample corpus is made of labelled NPs each containing a proper name of a known semantic class. Limiting the context of a proper name to the NP in which it appears is a trade-off between taking into account a window of an arbitrary size around the proper name (usually 1 or 2 words) and taking into account the entire sentence. The first prevents to model some collocations which might be relevant for semantic disambiguation while the second introduces too much noise in the automatic learning process.

The sample corpus is automatically built in three steps :

- The training corpus is tagged using the statistical tagger presented in section 2. All proper names belonging to the tagger's lexicon are automatically tagged according to equation 1.
- Then, this tagged corpus is parsed with a NP finite-state parser in order to detect NPs.
- Finally, the NPs containing proper names are stored in the sample corpus and tagged with the semantic class of the embedded proper name.

For example, the sentence : "the president of SONY declared in an interview ..", will first be tagged as :

```
(the,DETS)(president,NS)(of,PREP)(SONY,ORG)
(declared,V3S)(in,PREP)(an,DETS)(interview,NS)
```

Then the finite-state parser will isolate the following NPs :

```
[(the,DETS)(president,NS)(of,PREP)(SONY,ORG)]
[(an,DETS)(interview,NS)]
```

among which the NP *the president of SONY* will be kept since it contains a proper name (*SONY*) recognized as an organization name (ORG). The NP is stored in the sample corpus under the format :

```
(the,DET)(president,N)(of,PREP)(XXXX,PN)=ORG
```

Where the proper name is replaced by the symbol *XXXX* and its tag becomes *PN*, for proper name. This is to indicate that a proper name labelled ORG has occurred in this context.

At the end of this process a set of samples for each class of proper names has been built and the NPs whose number of occurrences exceeds a given threshold, are kept. This set of samples constitutes the training corpus on which the decision tree will be built.

4.2 Set of questions

The original aspect of SCT is the way in which the set of possible questions is generated. These questions ask whether a sequence of words and POS tags matches a certain regular expression involving words, POS and gaps.

During the growing process of the SCT, each node of the tree is associated with a regular expression called the *Known Structure* (*KS*) and a set of samples containing all the NPs from the sample corpus which satisfy this regular expression. At the beginning of the growing process, the root of the tree is associated to the $KS < + >$ and to the entire training corpus. A *KS* also records the position of the last item that was introduced in it.

The *KS* of a node and the set L composed of the lexical entries of the lexicon and the tagset will give rise to several new regular expressions by replacing in *KS* a gap with elements of L. More precisely:

- each element i of L produces four different patterns: $\{i\}$,$\{+i\}$,$\{i+\}$,$\{+i+\}$
- each of the generated patterns replaces in *KS* the gap situated respectively at the right and at the left of the element of the last item introduced.

With this method, a given *KS* generates a maximum of $4 \times |L| \times 2$ regular expressions. A 2K word lexicon and a tagset containing 100 POS tags, will produce for each *KS*, $4 \times (2000 + 100) \times 2 = 16.8K$ different new regular expression. The *KS* : $< +president+ >$, and the lexical item *of* will produce the eight following regular expressions:

$$\begin{array}{cc} < of\ president+ > & < +of\ president+ > \\ < of + president+ > & < +of + president+ > \\ < +president\ of > & < +president + of > \\ < +president\ of+ > & < +president + of+ > \end{array}$$

Each regular expression splits the set of samples associated to the node in two: the set of the samples that match the regular expression and the set of those that don't. A regular expression is therefore seen as a yes-no question.

4.3 Split criteria

The choice of the regular expression that will be associated to a node, is made in accordance with the Gini impurity criteria (Breiman et al., 1984). The best question (here regular expression) is the one which brings the maximum drop in *impurity* between the node and its children. If the two children of a node T are called T_{yes} and T_{no}, the drop of impurity ΔI is defined as:

$$\Delta I = I(T) - \frac{|T_{yes}|}{|T|} I(T_{yes}) - \frac{|T_{no}|}{|T|} I(T_{no}) \quad (2)$$

Impurity of a node t with i and j ranging over the tagset is computed with following formula :

$$I(t) = \sum_{i \neq j} p(i|t) p(j|t) \quad (3)$$

where $p(i|t)$ is estimated with the relative frequency of samples labelled with tag i in t.

The question associated to node t will be the question which maximises ΔI.

4.4 Stop condition

A node of the tree is not further split if either the impurity of the node is below a threshold or the number of samples left in a node equals 1.

At the end of this training process a tree has been built. Each node is associated to a regular expression made with words, POS and gaps. Each leaf contains a set of samples from the training corpus : those that match the regular expression represented by the leaf. From this set, a probability distribution over the different proper names semantic classes is computed. For example, if a leaf contains 100 samples, 90 of which are labelled with the tag TOWN and 10 labelled with ORG, the class TOWN will receive the probability 0.9 and class ORG the probability 0.1. The more uniform this distribution is, the less representative of a semantic class the leaf regular expression will be.

The shape of a node distribution allows us to distinguish the leaves with respect to their ability to discriminate one semantic class, such leaves will be called *discriminant*. A leaf will be considered discriminant when the probability of its top category exceeds a certain threshold. The probability of the top category will be called *discriminance* of the leaf and the threshold will be called the *minimum discriminance threshold*. When the minimum discriminance threshold is set to 0, all leaves are considered discriminant.

4.5 Estimating the quality of an SCT

The quality of an SCT has been evaluated by testing its ability to correctly tag a proper name occurrence according to the NP in which it appears. A test corpus of labelled NPs has been gathered and each NP has been given a tag by the SCT. The tagging process is straightforward, it consists on traversing the tree, starting at the root until a leaf is reached. For each node N visited, if the NP matches the regular expression of N, the next node to visit is the daughter of N labelled *yes*, otherwise, it is the daughter labelled *no*. If the leaf reached is discriminant (its discriminance is above the minimum discriminance threshold), the embedded proper name is labelled with the top category. If the leaf is not discriminant, the proper name is not given any tag. Three figures have been computed:

- the precision, which is the number of proper name occurrences correctly tagged divided by the number of proper name occurrences tagged;
- the syntactic coverage, which is the proportion of NPs occurrences in the test corpus that match a discriminant leaf's regular expression;
- the lexical coverage, which is the number of different proper names tagged at least one time by the SCT divided by the total number of different proper names in the test corpus

The training corpus is composed of NPs extracted from the newspaper *Le Monde*, between years 1987 and 1991 which constitute a corpus of almost 98 M words and a vocabulary of almost 400 Kwords. This corpus has been processed following the different steps described in section 4.1. In order to limit spurious NPs due to wrong PP attachments, we have limited the coverage of the grammar to NPs containing, at most, one PP attachment. The length of the NPs actually detected range from 1 to 12 words.

We kept, in the sample corpus, all the NPs occurring at least four times. This represents a set of 107K different NPs. The regular expressions were built on the vocabulary made of the 105 tags of the tagset and the 11K words of the sample corpus. At the end of the growing process, a tree containing 21K nodes is obtained [1]. Here are some examples of the regular expressions attached to the leaves of the tree (the proper name is represented by the letter X) :

```
+président++administration de DET X+   =XSOC
+président PREP gouvernement de DET X+ =XPAY
+président PREP directoire de DET X +  =XSOC
+le+ +président  PREP +  + de DET X+   =XSOC
```

All these examples correspond to nodes where the discriminance value is maximal (equal to 1).

Experiments have been conducted on two different test corpora :

- C_0 is made of $1.2M$ NPs from the newspaper *Le Monde* for the years 1991 and 1992. Each NP contains a proper name appearing in the lexicon, representing 4.6K different proper names. This test corpus is therefore close to the training corpus (although they correspond to different years), the aim of the experiment is to test the ability of the SCT to model the training data.
- C_1 is made of 695 NPs containing 282 different proper names appearing no more than 4 times in the years 91 and 92 of the newspaper *Le Monde*. The aim of this experiment is to test the ability of the SCT to correctly tag low frequency proper names (We make the assumption that *real* OOV proper names are sparse).

The results of the experiment on corpus C_0 are reported in figure 2, this figure shows coverage and precision with respect to the minimum discriminance threshold. These results show that a good precision can be reached by raising the minimum discriminance threshold. Raising the minimum discriminance threshold tends on the other hand to lower coverage.

The three different measures allow to draw different conclusions on this experiment.

The shape of the precision curve indicates that contexts that have showed to be discriminant on the training corpus led to a correct tagging on the test corpus. The shape of the syntactic coverage curve shows on another hand that an important

[1] The SCT was built using a toolkit developped in the framework of the SMADA project funded by the European Commission (Boves et al., 2000).

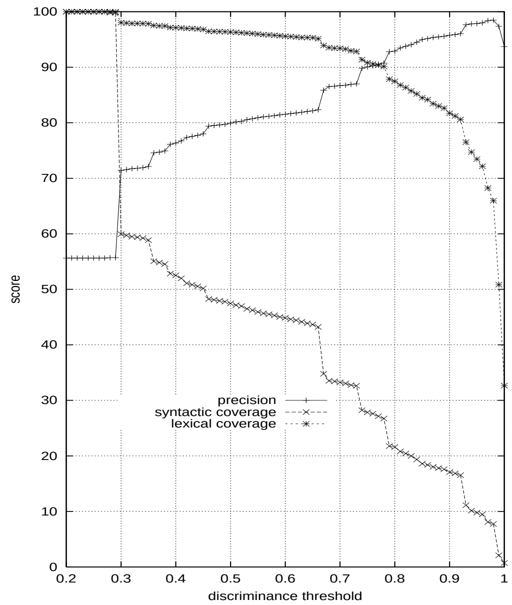

Figure 2: Recall and precision of an SCT with respect to the minimum discriminance level

part of the test corpus is composed of non discriminant contexts. We believe that this fact is due to two phenomena:

First, some contexts are truly ambiguous. For example, in the context *the president of X*, it is impossible to decide if X refers to a country name or an organization name. It is therefore inevitable that a proper name appearing in this context is sometimes mistagged by the SCT.

Second, restricting the context of a proper name to the NP in which it appears is sometimes too limited in order to correctly tag the proper name: some disambiguating elements may lie outside the NP. It would therfore be interesting to consider a larger context around a proper name or use a finer syntactic analysis, as done in (Collins and Singer, 1999) for named entity extraction. However, this will decrease the robustness of our method by being more sensitive to parsing errors due to ambiguity.

The lexical coverage curve shows that an important proportion of different proper names appear at least one time in a discriminant context. The idea that is explored in the following section consists in taking into account these occurrences of a proper name in order to update its entry in the lexicon.

The results of the experiment on corpus C_1 are very close to the results shown in figure 2 except for the lexical coverage which is relatively close to the syntactic coverage (most of the proper names appear only once in the corpus). This fact shows that the regular expressions learned by the SCT seem to model the context of a *class* of proper names or, put differently, that proper names of a given class tend to appear in similar contexts.

5 Updating a lexicon using a SCT

The previous experiments showed that the SCT is able to correctly tag a proper name with a high accuracy when the latter appears in a *low ambiguity* NP, which is an NP that matches a high discriminance leave's regular expression. We decided to take advantage of this property end select, in the test corpus, a subset of low ambiguity NPs containing the unknown proper names we want to include in the lexicon. This set of NPs is used to update the lexical entries of the proper names. The lexicon will itself be used to estimate the parameters of a statistical tagger. Recall that the lexica assign to every word w a count distribution over the different tags of the tagger tagset. The count distributions are updated in the following way:

- First, a minimum discriminance threshold S_d is chosen.

- Then, all the NPs of the test corpus in which appears a given unknown word w are processed through the tree and end up in different leaves[2].

- If the discriminance of the leaf is below S_d, the NP is rejected, otherwise the probability distribution of the leaf is taken into account to update the count distribution associated to w. For example, if w ends up in a leaf which gives a probability of 0.6 to the tag ORG and 0.4 to the tag COUNTRY, the counter of tag ORG in w's lexical entry is credited with 6 units and the counter of COUNTRY with 4 units, as if w was seen 6 times tagged ORG and 4 times tagged COUNTRY.

- Once all the NPs are processed, an estimate of the probability distribution $P(w_i|t_i)$ of equation 1 is computed using the count distributions of the lexicon.

[2]Contrary to the training corpus, in the test corpus, the category of the unknown words is unknown ! they are tagged UNK. In order to extract NPs, the parsing grammar was modified: all proper name symbols (ORG, FIRST ...)in the grammar were replaced by the symbol UNK. This technique introduces some noise by recognizing some spurious NPs. The importance of this noise has not been quantified.

This method relies on the implicit assumption that a proper name appearing in an ambiguous context tends to be ambiguous and that its lexical distribution reflects the probability distribution (as estimated by the SCT) of the context in which it appears. This assumption is of course too strong and non ambiguous words can appear in an ambiguous context. The proper name *Lebanon*, for example is not very ambiguous while the context *the president of* X, in which *Lebanon* can appear, is. Estimating the ambiguity of *Lebanon* with the ambiguity associated to the context *the president of* X is not very convincing and the assumption introduces a bias in the method. The importance of this bias is nevertheless attenuated by the fact that an OOV word can appear in several different contexts, in which case, its lexical distribution, as estimated by the SCT, will be less dependent of one context. The bias can be further attenuated by raising the parameter S_d, and therefore take into account low ambiguity contexts only in estimating the lexical distribution of a proper name.

5.1 Experiments

The approach described in section 5 has been compared to a standard technique for handling OOV words in POS tagging, which constitutes a baseline model. Experiments conducted using both techniques are reported in the following two sections. They were done on the 47.6 Kwords test corpus T made of the sentences from which were extracted the NPs of corpus C_1, presented in 4.5. The experiments consisted on considering that the elements of set E, made of the 282 proper names of C_1, were unknown to the lexicon and on estimating their lexical entries using two different techniques. After the new lexical distribution were computed, T was tagged again, yielding the tagged corpora T_{base} and T_{SCT} and the accuracy of the tagging computed.

5.1.1 Baseline model

In the baseline model, an assumption is made that all elements of E share the same uniform lexical distribution. This is a standard technique for handling OOV words in POS tagging (Weischedel et al., 1993). When the tagger is faced with an OOV word in a sentence, its most probable POS tag is chosen with respect to the sole trigram probability. The accuracy on T_{base} reached 67.3%, meaning that 67.30% of the occurrences of the words of E were given the right tag. This result is consistent (although a bit higher, due the difference of the tagsets size) with the equivalent experiments in (Weischedel et al., 1993).

5.1.2 SCT model

In this experiment, the lexical distribution of the elements of E are estimated following the process described in 5, using corpus T. The results obtained strongly rely on the discriminance threshold S_d chosen. For a given S_d, only a subset E' of E received new lexical distribution, the other items remained with their uniform distributions. Of course, if S_d is set to 0, then $E' = E$. Tagging accuracy of T_{SCT} for different values of S_d are reported in table 1 which also shows the relative gain compared to the baseline model. The best result (73%) is obtained with S_d set to 0.4.

S_d	0.0	0.2	0.4	0.6	0.8	1
T_{SCT}	71.3	71.3	73.0	72.31	70.8	67.5
$\%gain$	5.9	5.9	8.5	7.5	5.2	0.3

Table 1: Accuracy of T_{SCT} limited to set E

The drop of accuracy observed for values of S_d higher than 0.5 can be explained by the decrease of the size of E' when S_d increases, as shown by the $\frac{|E'|}{|E|}$ curve in figure 3. Figure 3 displays also accuracy of T_{base} and T_{SCT} limited to the items of E'. These two curves allow to compare more finely the performances of the two tagging techniques (uniform distribution *v/s* SCT estimated distribution) since the accuracy is computed only on items of E' which, by definition of E', are tagged using different distributions in T_{base} and in T_{SCT}.

The shapes of these two curves call for two comments. The curve $T_{SCT}(E')$ is always higher than $T_{base}(E')$, meaning that SCT estimated lexical distributions are always better than uniform distributions. The average absolute gain of $T_{SCT}(E')$ over $T_{base}(E')$ reaches 9%. The curve $T_{base}(E')$ shows an improvement of accuracy (from 67.3% up to 88%) for the items of E'. This fact indicates that when a context has been judged discriminant according to the SCT, it is also more discriminant for the trigram model alone.

6 Related work

Within the framework of Named Entity Extraction, popularized by the MUC conferences, several methods have been proposed to handle OOV proper names. In most of these works it is difficult to distinguish the process of unknown words tagging from the process of named entity detection. These methods can be classified as follows : hand-coded rules (which can also be associated with statistical methods), co-occurrences between words, Maximum-Entropy, Decision Trees and Hidden Markov models.

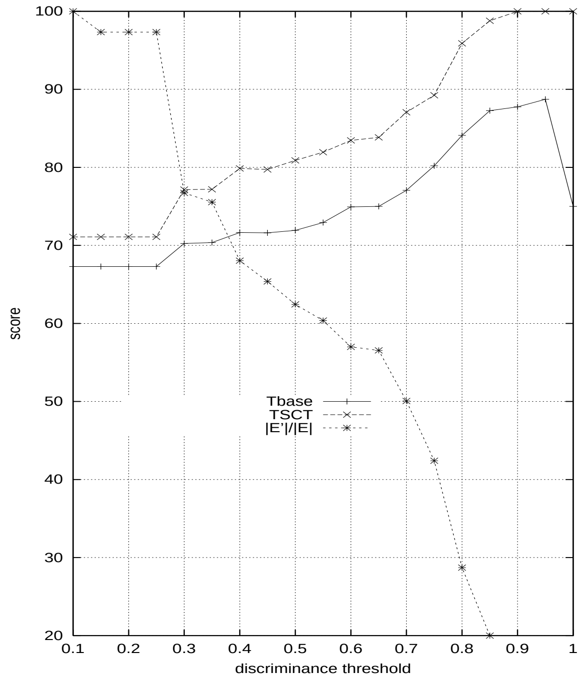

Figure 3: Accuracy of T_{base} and T_{SCT} restricted to E'

A set of predefined features is used by all these methods in order to classify proper names or named entity. These features can be a list of keywords, a position within the noun phrase including the proper name or some typographic informations as capitalization or format of numerical expressions. These features are used either to build hand-coded rules or as parameters in statistical approaches.

Compared to all the other methods listed above, our approach has one main originality: we don't use a set of predefined features during our learning process. In fact, any item of our training corpus (words and POS) can be used to model a particular context. It is the split criteria, used at each node of the tree which chooses an item (word or POS) and a position within the regular expression to make up a question.

It is important to note that our method can easily be combined with other statistical method developped for the name entity classification task. For example, in the system *IdentiFinder*^{TM} (Bikel et al., 1999), all the unknown proper names are grouped in a unique category labelled UNK whereas our method spreads them over finer classes. The lexicon produced by our method can directly be used in the Markov model of the system in order to improve the precision for such words.

A comparison of our results with other systems is not straightforward, because, as it was said before, the tagging precision of unknown proper names is not considered, most of the time, as a subtask of the named entity classification task. Nevertheless, the following papers give some results which can be compared to ours :

(Wascholder et al., 1997) carried on an experiment on a disambiguation task with three semantic tags (PERSON, PLACE, ORG). The test corpus was made of 1354 proper name tokens extracted from 88 Wall Street Journal documents. The tagging accurry reported is 82%.

(Collins and Singer, 1999) presents a method using only a very limited set of *seed* rules in order to automatically learn, from an unlabelled corpus, contexts relevant to disambiguate words belonging to the same set of tags as the previous work presented. The best accuracy reported, on a test corpus of 1000 contexts picked at random on the training data (New York Times text corpus), is 91.3% without taking into account the errors due to a wrong context detection and 83.3% with all the contexts detected.

We believe that the performance of our method is quite comparable to the score given above, according to the fact that our tagset is more precise (FAMILY and FIRST for PERSON; TOWN and COUNTRY for PLACE). Besides, including in the test corpus only items occuring less than four times makes the task more difficult than picking then at random.

7 Perspectives

As it is shown in section 5.1, this method works well when the unknown proper names occur, at least once, in a low ambiguity context in the test corpus: in figure 3, the tagging accuracy reaches 95% for words appearing at least once in a context with a discriminance higher than 0.8. It is therefore natural to try to obtain, for a given unknown proper name, as many context of occurrence as possible. This should increase the probability of finding discriminant contexts for characterizing it.

We started studying a method where the World-Wide-Web (WWW) is probed for more samples when the test corpus does not contain enough occurrences of a given unknown proper name to update its lexical entry.

The main problem when looking for new samples on the WWW is the noise associated with the answer to a query. The answers need to be processed in order to constitute valid samples. This processing involves cleaning and filtering after data has been sent back by a search engine.

Such processing involves the following steps :

1. Sending a query to a search engine for each unknown proper name we want to process. The two parameters of the query are the proper name as a keyword and the language of the text.

2. From all the answers sent back by the engine, only textual data is kept (HTML files). The HTML tags are then removed and the text is tokenised, tagged and parsed like the training corpus used for the building of our tree. Eventually, the NPs containing the proper name are kept.

3. All these NPs are processed through the tree and only those ending up in discriminant leaves are kept.

At the end of this process these samples are added to the samples already appearing in the test corpus to reestimate the POS tagger model.

We carried out a first experiment on the set E presented in 5.1. Each item of E, processed by the method previously described, generates on average a 3Mo HTML text corpus. After the cleaning process, only 110Ko of HTML text is kept for each item. Then, a corpus C_2 containing the 695 NPs of C_1 in addition to the 5K NPs extracted from the HTML corpus is built. After updating the lexicon E following the method described in 5 with C_2, the test corpus is tagged, yielding T_2. The tagging accuracy is reported in table 2: even if a slight improvement is observed, with respect to corpus T_1, the result is quite disapointing.

S_d	0.0	0.2	0.4	0.6	0.8	1
T_1	71.3	71.3	73.0	72.31	70.8	67.5
T_2	72.6	72.6	73.5	73.7	73.9	70.9

Table 2: Results T_1 and T_2 on the set E

A manual check on the data collected on the WWW gave us some clues in order to explain this poor improvement: first, even with the cleaning process, the data collected remain very noisy (lack of punctuation, HTML tag errors, wrong sentence and word tokenization, etc.); second, the differences between the data used to train the tree (newspaper articles) and those found on the WWW (litterature, chat, etc.) leads the SCT to misclassify proper names, even with a high discriminance threshold.

Nevertheless, this first experiment encourage us to carry on this way. Probing the WWW in order to automatically update semantic lexicon for proper names is a promising approach since proper names appear and disapear every day and we believe that an automatic searching and learning process can help producing relevant resources which can be directly used in any statistical NLP application.

References

Frédéric Béchet and François Yvon. 2000. Les noms propres en traitement automatique de la parole. *to appear in Traitement Automatique des Langues.*

Daniel Bikel, Richard Schwartz, and Ralph Weischedel. 1999. An algorithm that learns what's in a name. *Machine Learning - Special Issue on NL Learning*, 34,1-3.

Lou Boves, Denis Jouvet, Juergen Sienel, Renato de Mori, Frédéric Béchet, Luciano Fissore, and Pietro Laface. 2000. Asr for automatic directory assistance : the smada project. In *to appear in ISCA workshop : ASR2000*, Paris.

L Breiman, J Friedman, R Ohlsen, and C Stone. 1984. *Classification and Regression Trees.* Wadsworth.

Eugene Charniak, Curtis Hendrickson, Neil Jacobson, and Mike Perkowitz. 1993. Equations for part-of-speech tagging. In *11th National Conference on Artificial Intelligence*, pages 784–789.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *Empirical Methods in NLP processing and Very Large Corpora - EMNLP-VLC'99*, University of Maryland.

Roland Kuhn and Renato de Mori. 1996. The application of semantic classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):449–460, May.

Thierry Spriet and Marc El-bèze. 1995. Etiquetage probabiliste et contraintes syntaxiques. *Traitement Automatique des Langues*, Vol 36, n1-2.

Nina Wascholder, Yael Ravin, and Misook Choi. 1997. Disambiguation of proper names in text. In *Applied Natural Language Processing, ANLP'97.*

Ralph Weischedel, Richard Schwartz, Jeff Palmucci, Marie Meteer, and Lance Ramshaw. 1993. Coping with ambiguity and unknown words through probabilistic models. *Computational Linguistics*, 19(2):359–382.