Variable-Length Class Sequences Based on a Hierarchical Approach: MCnv

Imed Zitouni, Kamel Smaïli, Jean-Paul Haton

VARIABLE-LENGTH CLASS SEQUENCES BASED ON A HIERARCHICAL APPROACH: MC_n^ν

Imed ZITOUNI, Kamel SMAILI, and Jean-Paul HATON
LORIA / INRIA-Lorraine
Campus Scientifique B.P.239
F-54500 Vandœuvre-lès-Nancy, France

ABSTRACT

In this paper, we describe a new language model based on dependent word sequences organized in multi-level hierarchy. We call this model MC_n^ν, where n is the maximum number of words in a sequence and ν is the maximum number of levels. The originality of this model is its capability to take into account dependent variable-length sequences for very large vocabulary. In order to discover the variable-length sequences and to build the hierarchy, we use a set of 233 syntactic classes extracted from the eight French elementary grammatical classes. The MC_n^ν model learns hierarchical word patterns and uses them to reevaluate and filter the n-best utterance hypotheses outputed by our speech recognizer MAUD. The model have been trained on a corpus (LeM) of 43 million of words extracted from "Le Monde" a French newspapers and uses a vocabulary of 20000 words. Tests have been conducted on 300 sentences. Results achieved 17% decrease in perplexity compared to an interpolated class trigram model. Rescoring the original n-best hypotheses results also in an improvement of 5% in accuracy.

Keywords: speech recognition, language model, n-gram, class, syntactic classes, word sequences, hierarchic model.

1 INTRODUCTION

The use of statistical language models is a well known mean for introducing additional constraints in a speech recognizer and hence improving its performance. The role of the language model is to estimate the prior probability of the word under recognition. Most current language models are based on n-gram or on their variants where the probability of a word is made dependent on the past $n-1$ words of its history. However, these models achieve the boundary of their performances for n values around 3 or 4. Consequently, this kind of models doesn't take into account long distant constraints. In the following we will present a new approach for language modeling based on variable length sequences which is integrated in a real 20K speech recognizer. This model is based on an hierarchy of variable syntactic sequences classes. Theses classes are those which are assigned to the sentence words to be evaluated. This new approach is a generalization of the concept of multigram [1] as we will explain it further. In the following of the article, we give a brief overview of the evolution of language models and we discuss the manner of tagging each word of a sentence with its syntactic class according to the context (Section: 3). Next, we present the MC_n^ν model (Section: 4). A formulation of the model is given (Section: 5). We then report an evaluation of the MC_n^ν model and a comparison with the multiclass and interpolated class n-gram model (Section: 6). Finally, we conclude and give some perspectives.

2 OVERVIEW

In this section we approach the evolution of language models in order to introduce the one we propose in this paper. A refinement of the classical n-gram models is the class n-gram (n-class) model, where words are partitioned into equivalence classes (manually or automatically determined), and the inter-word transition probability is assumed to depend only on the word classes [7, 14]. This model copes with the sparseness data problem which is very crucial in the statistical estimation of parameters. The limit of this model is due to the fixed window of the history.

Another model, named multigram, has been developed in [1]. This model takes into account variable sequences of words in the history of each word under processing [1, 2]. In fact, the base line version of multigram models a sentence as a stream of independent sequences of words. Ideally, the structure of these word sequences corresponds to a syntactic units or phrases of variable length, as noun phrases, prepositional phrases, verb phrases... Therefore, the independent assumption between word sequences, supposed by the multigram model, contradicts the language structure. First works on modeling dependencies in [4], which combine multigram and bigram approaches, result in a large increase of the number of parameters and give less good performances than a trigram model. In plus, due to the data sparseness and due to the large number of parameters needed by this approach, as well as the multigram, these language models are unusable with a large vocabulary.

Motivated by the success of class based approaches in traditional n-gram modeling to solve the problem of data sparseness, we explored their potential in multigram. In plus, the introduction of syntactic classes in multigram approach allows to better take into account the linguistic dependencies between words. For that, we constructed a class based multigram from a suitable tagged corpus. This approach, called multiclass, is able to use large vocabulary which is not the case of the multigram model. Thus, the multiclass approach models a sentence as a stream of independent word sequences according to their syntactic classes [18].

However, this approach still suffers from the sequence independence assumption of classical multigram. In the multiclass concept, the dependency between words is taken into account inside a sequence, and there is no relationship between sequences. The MC_n^ν approach we propose is to overcome this independence assumption by building an hierarchy according to variable-length sequences of syntactic classes.

Language models based on the hierarchical principle have

been employed in other research. In particular, the use of a probabilistic finite state grammars reported by Hu *et al.* [9] as well as the use of n-gram reported by Jang *et al.* [10] to build the hierarchy of a sentence.

3 TAGGING A SENTENCE

The concept of class is very important in the model presented below. We discuss here the set of classification and the manner to tag each word of a sentence with its corresponding syntactic class. One way to formulate the problem is as follows: given a sentence $W(w_1 w_2 \dots w_n)$ how to determine the syntactic categories $C(c_1 c_2 \dots c_n)$ that maximize:

$$P(c_1 \dots c_n \,/\, w_1 \dots w_n) = \frac{P(c_1 \dots c_n)\, P(w_1 \dots w_n \,/\, c_1 \dots c_n)}{P(w_1 \dots w_n)}. \quad (1)$$

As we are interested in finding $c_1 c_2 \dots c_n$, the denominator will not affect the computation. By making some independent assumptions and bringing the model to a 3-class, formula (1) can be expressed as:

$$P(c_1 \dots c_n / w_1 \dots w_n) = P(c_1)P(w_1/c_1)P(c_1/c_2)P(w_2/c_2) \times \prod_{i=3}^{n} P(c_i/c_{i-2}c_{i-1})P(w_i/c_i). \quad (2)$$

In order to estimate the probabilities $P(c_i/c_{i-2}c_{i-1})$ and $P(w_i/c_i)$, we need to tag each word of the training corpus. Consequently, the dictionary of the application needs a syntactic field for each entry. This involves that some words have to be duplicated if they appear in more than one class. From the 8 elementary grammatical classes of French, we built up 233 syntactic classes, including punctuation [15]. These classes are divided into two groups: the opened and closed classes. A closed class is made up of a finite number of words (such as articles, preposition, ...). An open class is made up of words which can be formed from root's word (such as verbs, nouns, ...), or from personal nouns. Each unknown word in a sentence is supposed to belong at one of these opened classes and each punctuation symbol is in a single class.

The probability $P(c_i/c_{i-2}c_{i-1})$ can be expressed as a relative frequency

$$P(c_i/c_{i-2}c_{i-1}) = \frac{n(c_{i-2}c_{i-1}c_i)}{n(c_{i-2}c_{i-1})} \quad (3)$$

where $n(x)$ counts the number of times that the syntactic structure x occurs in a training text. In practice, this probability is obviously interpolated. One of the first steps consists of collecting the counts of 3-class (a sequence of 3 classes) and 2-class (a sequence of 2 classes). For that purpose we labeled a small text by hand, and with the statistics collected we tagged automatically a text of 0.5 million of words extracted from L'*Est Républicain* French newspaper. This tagging has been checked by hand, and the automatic labeling errors have been corrected. After that, we labeled automatically a corpus of 43 million words which represent 2 years (1987-1988) of *Le Monde (LeM)* newspaper. Tagging a corpus means to find the most likely sequence of syntactic classes for a sequence of words. In our approach we used a modified Viterbi algorithm [16].

The probability $P(w_i/c_i)$ is expressed as follow:

$$P(w_i/c_i) = \frac{n(w_i/c_i)}{n(c_i)} \quad (4)$$

where $n(w_i/c_i)$ counts the number of times that the word w_i is tagged by c_i in the training corpus.

4 THE MC_n^ν MODEL

The basic n-multiclass[1] language model is a kind of a generalization of the n-multigram model described in [2]. The n-multiclass is based on the same principles as the n-multigram with the difference that classes are used instead of words. In this language model, we assume that a sentence is considered as the concatenation of independent variable-length sequences of words. These variable-length sequences are built according to the syntactic class of each word in the sentence.

In the approach we propose, we begin by tagging each word of a sentence by its corresponding syntactic class. Then, we use a hierarchical approach to model dependencies between them. Indeed, the syntactic class phrase, corresponding to the sentence, is modeled by the concatenation of dependent variable-length class sequences (we hope that these variable-length sequences coincide with those defined traditionally in natural language, as noun phrases, prepositional phrases or verb phrases). The dependence between class sequences is carried out according to a certain hierarchy. For feasible modeling, we must specify the maximum number of syntactic classes in a class sequence, as well as the depth of the hierarchical model. We denote a model having maximum length n and depth ν as MC_n^ν. Using this notation, the traditional multiclass can be written as MC_n^1.

The MC_n^ν model proceeds as follows: first, we tag each word of the sentence by its syntactic class, according to the context, building a class phrase (level 0). After, at each level j ($j \in \{1 \dots \nu\}$) of the hierarchy, we build the best segmentation of the class phrase of level $(j-1)$ (§section 5), obtaining a class sequence phrase. Each class sequence of this segmentation became a class, building the class phrase of the upper level j. This process is repeated until $j = \nu$. The probability of a sentence is computed according to the tagging likelihood and the likelihood of the class sequence phrase obtained at level ν. The best segmentation of a class phrase is that having the greater likelihood. The likelihood of a class sequence phrase is the probability product of class sequences which compose it (§formula 7).

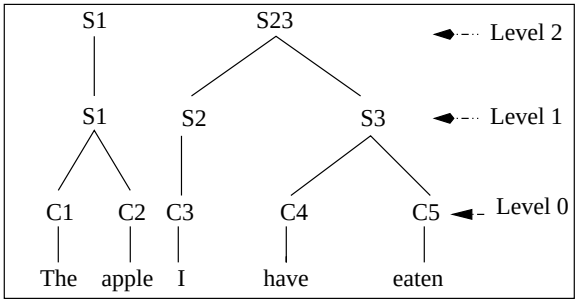

Figure 1: MC_3^2 on the sentence: *"The apple I have eaten"*

In figure 1 we present an example of applying the MC_3^2 to the sentence: "The apple I have eaten". In this example, the number of hierarchy level is limited to 2 and the maximum number of classes in one sequence is equal to 3. After tagging the sentence with the class phrase c_1, c_2, c_3, c_4, c_5 (level 0), we build the better segmentation obtaining the class sequence phrase s_1, s_2, s_3 (level 1): s_1 denotes the class sequence

[1] class multigram such us the maximum number of classes in a sequence is equal to n

(c_1, c_2), s_2 denotes the class sequence c_3 and s_3 denotes the class sequence (c_4, c_5). Level 2 contains the best segmentation of the level 1 class phrase (s_1, s_2, s_3).

5 FORMULATION OF THE MC_n^ν MODEL

The likelihood of a sentence $\mathcal{W} = (w_1, w_2, \ldots, w_k)$ tagged by the syntactic class phrase $\mathcal{C} = (c_1, c_2, \ldots, c_k)$ is formulated as follows:

$$P(\mathcal{W}) = (\prod_{i=1}^{i=k} p(w_i/c_i)) \times P(\mathcal{C}) \tag{5}$$

where $p(w_i/c_i)$ denotes the probability that the word w_i is tagged by the class c_i (§formula 4) and $P(\mathcal{C})$ denotes the likelihood of the class phrase $\mathcal{C}$. This likelihood $P(\mathcal{C})$ is computed according to the set of most likely segmentations at each level, thus yielding the approximation:

$$P(\mathcal{C}) = P_{MC_n^\nu}(\Omega_\nu) \tag{6}$$

where Ω_ν is a class phrase corresponding to the most likely segmentation $\mathcal{S}^*_{MC_n^{(\nu-1)}}$ of the class phrase of level $\nu-1$ (§formula 9), $\Omega_{\nu-1}$. This class phrase $\Omega_{\nu-1}$ corresponds also to the most likely segmentation of $\Omega_{\nu-2}$ (recursively until Ω_1). Each class of Ω_j $(1 < j \leq \nu)$ corresponds to a class sequence in Ω_{j-1}. Ω_1 corresponds to the syntactic class phrase $\mathcal{C}$. If $\nu = 1$ the MC_n^ν model is similar to the basic n-multiclass model.

Let $\mathcal{S}_j$ be a possible segmentation of the class phrase Ω_j building $(s_j(1), s_j(2), \ldots, s_j(q_j))$, where $s_j(i)$ corresponds to a class sequence in Ω_j. The likelihood $P_{MC_n^\nu}(\Omega_j, \mathcal{S}_j)$ of the class phrase Ω_j associated with segmentation $\mathcal{S}_j$ is the probability product of the successive class sequences $s_j(i)$ $(i \in \{1 \ldots q_j\})$, each of them has a maximum length of n:

$$P_{MC_n^\nu}(\Omega_j, \mathcal{S}_j) = \prod_{t=1}^{t=q_j} p(s_j(t)) \tag{7}$$

Denoting as $\{\mathcal{S}_j\}$ the set of all possible segmentation of Ω_j into class sequences, the likelihood of Ω_j is:

$$P^*_{MC_n^j}(\Omega_j) = \max_{\mathcal{S}_j \in \{\mathcal{S}_j\}} P_{MC_n^\nu}(\Omega_j, \mathcal{S}_j) \tag{8}$$

where the most likely segmentation $\mathcal{S}^*_{MC_n^j}$ of Ω_j is :

$$L^*_{MC_n^j} = \arg P^*_{MC_n^j}(\Omega_j) = \Omega_{j+1}. \tag{9}$$

For instance, with a maximum number of class in a sequence equal to 3 $(n = 3)$ and with two levels hierarchy $(\nu = 2)$, the likelihood of the class phrase $\mathcal{C} = abcd$ $(P(\mathcal{C}))$ is computed in an increasing way. We denote sequence borders with brackets. For $j = 1$ $(\Omega_1 = \mathcal{C} = abcd)$:

$$P^*_{MC_3^1}(\Omega_1) = max \left\{ \begin{array}{l} p([a])p([bcd]) \\ p([abc])p([d]) \\ p([ab])p([cd]) \\ p([ab])p([c])p([d]) \\ p([a])p([bc])p([d]) \\ p([a])p([b])p([cd]) \\ p([a])p([b])p([c])p([d]) \end{array} \right\}$$

Assume that $P^*_{MC_3^1}(\Omega_1) = p([a])p([bc])p([d])$ and let X being the new tag of the class sequence $[bc]$ $(X \equiv [bc]) : \Omega_2 = aXd$ and

$$P(C) = P^*_{MC_3^2}(\Omega_2) = max \left\{ \begin{array}{l} p([a])p([Xd]) \\ p([aX]p([d]) \\ p([aXd]) \\ p([a])p([X])p([d]) \end{array} \right\}$$

Assume that a training corpus W is tagged by the syntactic class corpus C. The model is thus defined by the optimal level of hierarchy ν and by the set of parameters Θ_j, $1 \leq j \leq \nu$, consisting of the probability of each sequence $s_j(i)$ in the dictionary of level j $D_{\mathcal{S}_j} : \Theta_j = \{p(s_j(i))\}$, with $\sum_{s_j(i) \in D_{\mathcal{S}_j}} p(s_j(i)) = 1$. $D_{\mathcal{S}_j} = \{s_j(i)\}$ denotes a dictionary of class sequences which can be formed by combining $1, 2 \ldots$ up to n classes from the training class corpus of level j (O_j). The most likely segmentation $\mathcal{S}^*_{MC_n^j}$ of the training class corpus O_j, allows to build the training class corpus of level $j+1$ (O_{j+1}) which is used to estimate the set of parameters Θ_{j+1}. Θ_1 is estimated on the training class corpus $C = O_1$.

Thus, we begin with the class corpus $C = O_1$ and we use the process described in the subsection 5.1 to extract sequences and to estimate the set of parameters Θ_1. Then, we build the most likely segmentation $\mathcal{S}^*_{MC_n^1}$ of O_1 (§formula 9), obtaining the second level training corpus O_2 with a likelihood equal to $P^*_{MC_n^1}(O_1)$. The class corpus O_2 is used to extract sequences and to estimate the set of parameters Θ_2 of the second level. We repeat this process at each level j until the corpus likelihood $P^*_{MC_n^j}(O_j)$ stop increasing:

$$P^*_{MC_n^j}(O_j) \leq P^*_{MC_n^{j-1}}(O_{j-1}),$$

obtaining the optimal level of hierarchy ν.

5.1 Maximum Likelihood Estimation of the Model Parameters

As mentioned above, O_j is the training class corpus at level j. This training class corpus is obtained with the most likely segmentation of the training class corpus of level $j-1$ (O_{j-1}). The MC_n^ν language model is completely defined by a set of parameters Θ_j $(j \in \{1 \ldots \nu\})$ consisting of the probability of each sequence $s_j(i)$ in a dictionary $(D_{\mathcal{S}_j} = \{s_j(i)\}_{i=1}^{i=m})$. Each sequence $s_j(i)$ can contains until n classes from O_j.

$$\Theta_j = \{p(s_j(i))\}_{i=1}^{i=m} \quad \text{where} \quad \textstyle\sum_{i=1}^{m} p(s_j(i)) = 1$$

The re-estimation formula of the set of parameters Θ_j can be obtained by the Maximum Likelihood (ML) estimation from incomplete data [5], where the observed data is the string of symbols O_j, and the unknown data is the underlying segmentation $\mathcal{S}_j$. Thus, iterative ML estimates of Θ_j can be computed through an EM algorithm.

Let $Q(k, k+1)$ be the following auxiliary function computed with the likelihoods of iterations k and $k+1$:

$$Q(k, k+1) = \sum_{\mathcal{S}_j \in \{\mathcal{S}_j\}} P^{(k)}(O_j,\ \mathcal{S}_j) \log P^{(k+1)}(O_j,\ \mathcal{S}_j) \tag{10}$$

Dempster et al. in [5] show that if $Q(k, k+1) \geq Q(k, k)$, then $P^{(k+1)}(O_j) \geq P^{(k)}(O_j)$. The set of parameters which

maximizes $Q(k, k+1)$ at iteration $(k+1)$ also leads to an increase of the corpus likelihood. Therefore the re-estimation formula of the parameters of iteration $(k+1)$, i.e., the probability of sequences $\{s_j(i)\}_{i=1}^{i=m}$, can be derived by maximizing the auxiliary function $Q(k, k+1)$.

Let $c(s_j(i), \mathcal{S}_j)$ denotes the number of occurrences of the sequence $s_j(i)$ in a segmentation $\mathcal{S}_j$ of the corpus at Level j. We rewrite the joint likelihood given in (7) so as to group together the probabilities of all identical sequences:

$$P^{(k+1)}(O_j, \mathcal{S}_j) = \prod_{i=1}^{i=m} (p^{(k+1)}(s_j(i)))^{c(s_j(i), \mathcal{S}_j)} \quad (11)$$

The auxiliary function $Q(k, k+1)$ can then be expressed as:

$$Q(k, k+1) = \sum_{i=1}^{m} \sum_{\mathcal{S}_j \in \{\mathcal{S}_j\}} P^{(k)}(O_j, \mathcal{S}_j)\, c(s_j(i), \mathcal{S}_j)\, \log p^{(k+1)}(s_j(i)). \quad (12)$$

This function is subject to the following constraints $\sum_{i=1}^{m} p^{(k+1)}(s_j(i)) = 1$ and $p^{(k+1)}(s_j(i)) \geq 0$. It reaches its maximum for [3]:

$$p^{(k+1)}(s_j(i)) = \frac{\sum_{\mathcal{S}_j \in \{\mathcal{S}_j\}} c(s_j(i), \mathcal{S}_j) \times P^{(k)}(O_j,\ \mathcal{S}_j)}{\sum_{\mathcal{S}_j \in \{\mathcal{S}_j\}} c(\mathcal{S}_j) \times P^{(k)}(O_j,\ \mathcal{S}_j)} \quad (13)$$

where $c(\mathcal{S}_j) = \sum_{i=1}^{i=m} c(s_j(i), \mathcal{S}_j)$ is the total number of sequences in $\mathcal{S}_j$. Formula (13) shows that the estimation of $p(s_j(i))$ is merely a weighted average depending on the occurrences of sequence $s_j(i)$ within each possible segmentation. Since each iteration improves the model in the sense of increasing the data likelihood $P^{(k)}(O_j)$, it eventually converges to a critical point.

The forward-backward algorithm, described in [2], can be used to reestimate the formula (13). The set of parameters Θ_j can be initialized with the relative frequencies of all co-occurrences of symbols up to length n in the training corpus. Then Θ_j is iteratively re-estimated until the training set likelihood does not increase significantly, or until a fixed number of iterations is reached [17].

Some pruning techniques may be advantageously applied in practice to the dictionary of sequences, in order to avoid overlearning. A straightforward way to proceed consists of simply discarding, at each iteration, the most unlikely sequences, i.e., those with probability values falling under a specified threshold.

6 EVALUATION

In this section, we assess the MC_n^ν model in the framework of language modeling with the training method described in subsection 5.1. Performances are evaluated in terms of test perplexity [11] and in terms of accuracy reported to our speech recognizer MAUD. We give also a comparison of the MC_n^ν model with the class n-gram and classical multiclass (MC_n^1) models.

6.1 Data Description

Models have been built on a French corpus (LeM) of 43 million of words which represents two years ($1987 - 1988$) of "Le Monde" newspaper. The class set used to evaluate our models contains 233 syntactic classes (conjugated verbs, infinitive verbs, defined articles, undefined articles, ...), including punctuation. These classes are extracted from the 8 elementary grammatical classes of the French language. The vocabulary used contains 20000 words provided by the AUPELF-UREF evaluation campaign (similar to the wall street journal DARPA test, but in French language) [6]. Five laboratories are participated, among whom three have got very good results in the DARPA campaigns. The base version of our speech recognizer MAUD, presented in subsection 6.3, is ranked second.

6.2 Perplexity Results

Perplexity is usually considered to be a performance measure of language models. It is therefore interesting to look at the test perplexity values obtained by the MC_n^ν approach at different levels of the hierarchy. It allows us to compare the performance yielded by this approach with a multiclass model and also with a classical interpolated class n-gram model (biclass and triclass). The perplexity of a test corpus $\mathcal{W}$ (T words) tagged by the class corpus $\mathcal{C}$ is computed as follows:

$$PP(\mathcal{W}) = 2^{-\frac{1}{T} log_2 P(\mathcal{W})} \quad (14)$$

where $P(\mathcal{W})$ is the likelihood of the test corpus. In the case of MC_n^ν and multiclass (MC_n^1) models, this likelihood value ($P(\mathcal{W})$) is computed according to the formula 5.

For MC_n^ν language model all co-occurrences symbols are used to get initial estimates of the sequence probabilities. However, to avoid overlearning, we found efficient to discard infrequent co-occurrences, i.e., those appearing strictly less than a given number of times C_0. This value of C_0 is determined experimentally on a corpus. In our experiments the best value of C_0 is equal to 8 ($C_0 = 8$). Then, ten training iterations are performed in this experiment at each level of the hierarchy (§subsection 5.1). Sequence probabilities falling under a threshold p_0 are set to 0, except those of length 1 which are assigned a minimum probability p_0. We set the fixed probability $p_0 \approx 5 \times 10^{-6}$ which is half the probability of a class occurring only once in the training corpus. After the initialization and for each iteration, probabilities are renormalized so that they add up to 1 [4]. Since all class sequences of length 1 have a minimum probability of p_0, the likelihood of any string of classes can be computed.

We show in figure 2 the test perplexity obtained by the MC_n^ν model for different values of n and ν. One can notice that figure 2 shows also the multiclass (MC_n^1) test perplexity for different values of n.

Experiments show that, progressively the number of hierarchy increases, the performances improve until a value of ν equal to 4 with a maximum number of classes in a sequence equal to 5 ($n = 5$). The test perplexity begin with a value equal to 145.31 for the MC_2^1 and decrease to 74.78 with a number of hierarchy equal to 4 and with sequence length equal to 5 (MC_5^4). The best performances of the multiclass model is given with a value of n equal to 7 ($PP = 93.52$).

Table 1 shows, for different sequence lengths n, the test perplexity of the multiclass (MC_n^1) and the MC_n^ν model with optimal level of hierarchy ($\nu = 4$).

The experiment concerning the interpolated class n-gram model, on the same corpus, gives a perplexity of 138.97 for the interpolated biclass model and 87.73 for the interpolated triclass model.

The perplexity comparison of multiclass (MC_n^1), MC_n^ν and interpolated class n-gram indicates that MC_n^ν outperform by 85% the performance of a biclass, by 17% the performance of a triclass and by 25% the performance of a multiclass model.

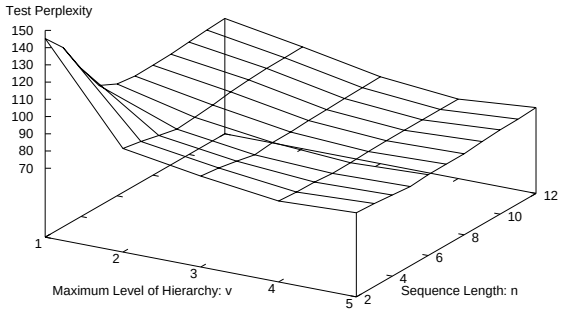

Figure 2: Test perplexity of the MC_n^ν model for different values of n and ν, where n denotes the sequence length and ν denotes the maximum level of the hierarchy.

n	4	5	6	7	8
$PP_{MC_n^1}$	$115,91$	$100,02$	$94,92$	$93,52$	$93,81$
$PP_{MC_n^4}$	$75,95$	$74,78$	$75,17$	$75,67$	75.94

Table 1: Test perplexity of the multiclass and the MC_n^4 model with different sequence lengths n.

6.3 MAUD System and Recognition Results

An evaluation was also done with MAUD [8], a continuous dictation system using a stochastic language model. Each phoneme is modeled by a second order hidden Markov model [13] with 3 states ($HMM2$). Thus, each word in the vocabulary is represented by the concatenation of the $HMM2$ phones which compose it. To estimate the HMM2s phones, we use Bref80 spoken corpus for French [12]. The basic version of MAUD works in 3 steps: gender identification, word lattice generation by means of a Viterbi block algorithm with a bigram model, N-best sentences extraction by means of a beam search in accordance with combined score of the acoustic and the trigram language models. The best sentence produced by the third step is the MAUD result.

To evaluate the performance brought by the introduction of our approach, we use the MC_n^ν model to rescore the N-best utterance hypotheses produced by the third step of MAUD: the best hypothesis after rescoring is the system result. We build also another version which use the multiclass, instead of the MC_n^ν model, to rescore the N-best hypotheses. Tests have been conducted on 300 French sentences. For each sentence, the number of hypotheses extracted from the third step is equal to 80 ($N = 80$).

The evaluation is done in terms of accuracy (**Acc**), word predicted correctly (**Corr**), substitution (**Sub**), deletion (**Del**) and insertion (**Ins**) rates. Table 2 gives this different rates for the basic version (**BV**), the version using multiclass ($\mathbf{MC_n^1V}$) and the one using MC_n^ν model ($\mathbf{MC_n^\nu V}$) with the optimal level of hierarchy ($\nu = 4$).

Results show that the version using the MC_n^ν (MC_n^νV) improves the accuracy by 2% compared to the one using the multiclass (MC_n^1V) and by 5% compared the basic one (BV).

	Acc	Corr	Sub	Del	Ins
BV	$55,2\%$	$61,7\%$	$29,7\%$	$8,6\%$	$6,5\%$
$\mathbf{MC_n^1V}$	$57,1\%$	$61,2\%$	$27,4\%$	$11,4\%$	$4,1\%$
$\mathbf{MC_n^\nu V}$	$58,0\%$	$62,7\%$	$27,5\%$	$9,8\%$	$4,7\%$

Table 2: Performances of different versions of MAUD system.

7 CONCLUSION AND PERSPECTIVES

We described in this paper a new language model which learns statistically hierarchical patterns of word phrases in spoken language utterances. This new model, able to use a large vocabulary, is used to rescore the N-best utterance hypotheses list which is outputed by a speech recognizer. The hierarchical approach models a sentence as a stream of dependent word sequences according to their syntactic classes. This dependence is according a hierarchy.

Experiments show that MC_n^ν could be a competitive alternative to the multiclass and interpolated class n-gram models. The MC_n^ν language model outperforms in terms of perplexity the interpolated biclass model by 85%, the triclass model by 17% and the multiclass approach (MC_n^1) by 25%. It outperforms also the accuracy of our large vocabulary continuous speech recognition system MAUD. In fact, the MAUD version which uses the MC_n^ν model outperforms the accuracy of the one using the multiclass by 2% and the base one by 5%. The base version is limited to the use of a n-gram language model.

We are investigating the application of the MC_n^ν approach to other issues, e.g, in looking for semantic equivalence classes between word sequences, in view of tagging concept and speech to speech automatic translation.

Acknowledgments

We want to thank Fréderic BIMBOT and Sabine DELIGNE for the package of multigram and for the faithful discussions we have about this new approach.

8 REFERENCES

[1] F. Bimbot, R. Pieraccini, E. Levin, and B. Atal. Modèles de séquences à horizon variable. In *Proc. XXth JEP*, Trégastel (France), June 94 1994.

[2] S. Deligne and F. Bimbot. Language modeling by variable length sequences : Theoretical formulation and evaluation of multigrams. In *Proceeding of the International Conference on Acoustics, Speech and Signal Processing*, pages 169–172, 1995.

[3] S. Deligne and F. Bimbot. Inference of variable-length linguistic and acoustic units by multigrams. In *Speech Communication*, volume 23, pages 223–241, 1997.

[4] S. Deligne, F. Yvon, and F.Bimbot. Introducing statistical dependencies and structural constraints in variable-length sequence models. In *Lecture notes in Artificial Intelligence 1147*, pages 156–167. Springer, 1996.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum-likelihood from incomplete data via the em algorithm. *J. Roy. Stat. Soc.*, 39(1):1–38, 1977.

[6] J.M. Dolmazon, F. Bimbot, G. Adda, M. El-Beze, J.C Caërou, J. Zeiliger, and M. Adda-Decker. Organisation de la première campagne aupelf pour l'évaluation des systèmes de dictée vocale. In *Actes des premières JST Francil 1997*, pages 13–18, Avignon, France, April 1997.

[7] F.Jelinek, R. Mercer, and S. Roukos. Principles of lexical language modeling for speech recognition. In Furui S. editor, editor, *Advances in Signal Processing*, pages 651–699. Marcel Dekker, 1992.

[8] D. Fohr, JP. Haton, JF. Mari, K. Smaili, and I. Zitouni. Maud: Un prototype de machine à dicter vocale. In *1 ères JST Francil 1997*, pages 25–30, Avignon-France, Avril 1997.

[9] J. Hu, W. Turin, and M.K. Brown. Language modeling using stochastic automata with variable length contexts. *Computer Speech and Language*, 11(1):1–16, January 1997.

[10] P.J. Jang and A. Hauptmann. Hierarchical cluster language modeling with statistical rule extraction for rescoring n-best hypotheses during speech decoding. In *International Conference on Spoken Language Processing*, Sydney, Australia, December 1998.

[11] F. Jelinek. Self-organized language modeling for speech recognition. In Alex Waibel and Kai-Fu Lee, editors, *Speech Recognition*, pages 450–506. Morgan Kaufmann, 1990.

[12] L. Lamel, J.L. Gauvain, and M. Eskenazi. Bref, a large vocabulary spoken corpus for french. In *Proceeding of European Conference on Speech Communication and Technology*, Gênes, 1991.

[13] J.-F. Mari, J.-P. Haton, and A. Kriouile. Automatic Word Recognition Based on Second-Order Hidden Markov Models. *IEEE Transactions on Speech and Audio Processing*, 5, January 1997.

[14] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependences in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

[15] K. Smaili, F. Charpillet, and JP. Haton. A new algorithm for word classification based on an improved simulated annealing technique. In *5th International Conference on the Cognitive Science of Natural Language Processing*, 1996.

[16] K. Smaili, I. Zitouni, F. Charpillet, and J-P. Haton. An hybrid language model for a continuous dictation prototype. In *EUROSPEECH97*, Rhodes (GREECE), September 1997.

[17] I. Zitouni. *Modélisation du langage pour les systèmes de reconnaissance de la parole destinés aux grands vocabulaires : application à MAUD*. PhD thesis, Université Henri Poincaré-Nancy 1, 2000.

[18] I. Zitouni, K. Smaili, J-P. Haton, S. Deligne, and F. Bimbot. A comparative study between polyclass and multiclass language models. In *Proceding of 5th International Conference on Spoken Language Processing*, 1998.