



## STATISTICAL LANGUAGE MODEL BASED ON A HIERARCHICAL APPROACH: $MC_n^\nu$

*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 propose a new language model based on dependent word sequences organized in a multi-level hierarchy. We call this model  $MC_n^\nu$ , where  $n$  is the maximum number of words in a sequence and  $\nu$  is the maximum number of levels. The originality of this model is its capacity to take into account dependent variable-length sequences for very large vocabularies. In order to discover the variable-length sequences and to build the hierarchy, we use a set of 233 syntactic classes extracted from the 8 French elementary grammatical classes. The  $MC_n^\nu$  model learns hierarchical word patterns and uses them to reevaluate and filter the  $n$ -best utterance hypotheses outputted by our speech recognizer MAUD. The model has been trained on a corpus of 43 million words extracted from a French newspaper 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 resulted in an improvement of 5% in accuracy.

### 1. INTRODUCTION

The most current language models are based on  $n$ -grams or on their variants, where the probability of a word is made dependent on the past  $n - 1$  words of its history. These models reach the limit of their performance for  $n$  values around 3 or 4. Consequently, such models do not take into account long distant constraints.

Another model, named  $n$ -multigrams takes into account variable sequences of words in the history of each word [1, 2]. The basic version of  $n$ -multigrams models a sentence as a stream of independent sequences of words. The maximum number of words in a sequence is equal to  $n$ . Ideally, the structure of these word sequences corresponds to syntactic units or phrases of variable length, such as noun phrases, prepositional phrases, verb phrases, etc. The independent assumption between word sequences, supposed by the  $n$ -multigrams model, thus contradicts the language structure. First works on modeling dependencies [3], combining  $n$ -multigrams and bigram approaches, resulted in a large increase of the number of parameters and gave less good performance than a trigram model. In addition, due to sparseness of data and to the large number of parameters needed, these language models are unusable with a large vocabulary.

Motivated by the success of class based approaches in traditional  $n$ -grams modeling to solve the problem of data sparseness, we have explored their potential in  $n$ -multigrams. Moreover, the introduction of syntactic classes in  $n$ -multigrams approaches allows us to better take into account the linguistic dependencies be-

tween words. For these reasons, we have designed the  $n$ -multiclass model which is able to use large vocabularies. The  $n$ -multiclass approach models a sentence as a stream of independent word sequences according to their syntactic classes [4]. This approach still suffers from the sequence independence assumption of classical  $n$ -multigrams. In the  $n$ -multiclass concept, the dependency between words is taken into account inside a sequence, and there is no relationship between sequences. The  $MC_n^\nu$  approach that we propose is intended to overcome this independence assumption by building a 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 was reported by Hu *et al.* [5] as well as the use of  $n$ -grams to build the hierarchy of a sentence [6, 7].

### 2. THE $MC_n^\nu$ MODEL

In the  $n$ -multiclass model, we assume that a sentence is the concatenation of independent variable-length sequences of words. These sequences are built according to the syntactic class of each word in the sentence.

In the approach we propose, we start by tagging each word of a sentence with its corresponding syntactic class [8]. Then, we use a hierarchical approach to model dependencies between words. The syntactic class phrase corresponding to the sentence is modeled by the concatenation of dependent variable-length class sequences (we hope that these sequences coincide with those defined traditionally in natural language, such as noun phrases, verb phrases, etc.). 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, and also the depth of the hierarchical model. We denote a model having maximum length  $n$  and depth  $\nu$  as  $MC_n^\nu$ . Using this notation, the traditional  $n$ -multiclass is a particular case of this model ( $\nu = 1$ ).

The  $MC_n^\nu$  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). Then, at each level  $j$  ( $j \in \{1 \dots \nu\}$ ) of the hierarchy, we look for the best segmentation of the class phrase of level  $(j - 1)$  (cf. §3), obtaining a class sequence phrase. Each class sequence of this segmentation becomes 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  $\nu$ . The best segmentation of a class phrase is the



one having the highest likelihood.

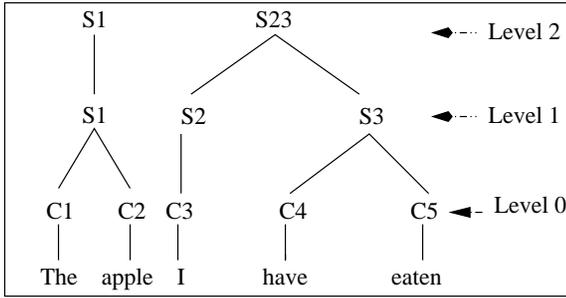


Fig. 1.  $MC_3^2$  on the sentence: “The apple I have eaten”

In figure 1 we present an example of applying  $MC_3^2$  to the sentence: “The apple I have eaten”. In this example, the number of hierarchy levels is limited to 2 and the maximum number of classes in one sequence is set 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 best segmentation obtaining the class sequence phrase  $s_1, s_2, s_3$  (level 1):  $s_1$  denotes the class sequence  $(c_1, c_2)$ ,  $s_2$  denotes the class sequence  $c_3$ , and  $s_3$  denotes the class sequence  $(c_4, c_5)$ . Level 2 contains symbols  $(s_1, s_2, s_3)$  assigned to each component of the best segmentation of level 1.

### 3. FORMULATION OF THE $MC_n^\nu$ MODEL

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

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

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

$$P(\mathcal{C}) = P_{MC_n^\nu}(\Omega_\nu) \quad (2)$$

where  $\Omega_\nu$  is a set of class phrases obtained from the most likely segmentation  $\Omega_{\nu-1} = \mathcal{S}_{MC_n^{\nu-1}}^*$  at level  $\nu - 1$ . This class phrase  $\Omega_{\nu-1}$  corresponds also to the most likely segmentation of  $\Omega_{\nu-2}$  (recursively until  $\Omega_1$ ). Each class of  $\Omega_j$  ( $1 < j \leq \nu$ ) corresponds to a class sequence in  $\Omega_{j-1}$ .  $\Omega_1$  corresponds to the syntactic class phrase  $\mathcal{C}$ . When  $\nu = 1$  the  $MC_n^\nu$  model is similar to the basic  $n$ -multiclass model.

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

$$P_{MC_n^\nu}(\Omega_j, \mathcal{S}_j) = \prod_{t=1}^{t=q_j} p(s_j(t)) \quad (3)$$

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

$$P_{MC_n^\nu}^*(\Omega_j) = \max_{\mathcal{S}_j \in \{\mathcal{S}_j\}} P_{MC_n^\nu}(\Omega_j, \mathcal{S}_j) \quad (4)$$

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

$$\mathcal{S}_{MC_n^j}^* = \arg P_{MC_n^j}^*(\Omega_j) = \Omega_{j+1}. \quad (5)$$

For instance, with a maximum number of classes ( $n$ ) in a sequence equal to 3 ( $n = 3$ ) and with a two level hierarchy ( $\nu = 2$ ), the likelihood of the class phrase  $\mathcal{C} = abcd$  ( $P(\mathcal{C})$ ) is computed in an increasing way. In the following, each segmented sequence is represented between brackets. For  $j = 1$  ( $\Omega_1 = \mathcal{C} = abcd$ ):

$$P_{MC_3^1}^*(\Omega_1) = \max \left\{ \begin{array}{l} p([a])p([bcd]) \\ p([abc])p([d]) \\ p([ab])p([cd]) \\ p([ab])p([c])p([d]) \\ p([a])p([bc])p([d]) \\ p([a])p([b])p([cd]) \\ p([a])p([b])p([c])p([d]) \end{array} \right\} \quad (6)$$

Assume that  $P_{MC_3^1}^*(\Omega_1) = p([a])p([bc])p([d])$  and let  $X$  be the new tag representing the class sequence  $[bc]$  ( $X \equiv [bc]$ ), then  $\Omega_2 = aXd$  and

$$P(\mathcal{C}) = P_{MC_3^2}^*(\Omega_2) = \max \left\{ \begin{array}{l} p([a])p([Xd]) \\ p([aX])p([d]) \\ p([aXd]) \\ p([a])p([X])p([d]) \end{array} \right\} \quad (7)$$

The model is thus defined by the optimal level of hierarchy  $\nu$  and by the set of parameters  $\Theta_j$ ,  $1 \leq j \leq \nu$ , consisting of the probability of each sequence  $s_j(i)$  in the dictionary of level  $j$   $D_{S_j} : \Theta_j = \{p(s_j(i))\}$ , with  $\sum_{s_j(i) \in D_{S_j}} p(s_j(i)) = 1$ . Actually,  $D_{S_j} = \{s_j(i)\}$  denotes a dictionary of class sequences which can be formed by combining  $1, 2 \dots$  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 us to build the training class corpus of level  $j + 1$  ( $O_{j+1}$ ) which is used to estimate the set of parameters  $\Theta_{j+1}$ .  $\Theta_1$  is estimated on the training class corpus  $C = O_1$ . We assume that the training corpus  $W$  is tagged by the syntactic class corpus  $C$ .

Thus, we begin with the class corpus  $C = O_1$  and we use the process described in the subsection 3.1 to extract sequences and to estimate the set of parameters  $\Theta_1$ . Then, we build the most likely segmentation  $\mathcal{S}_{MC_n^1}^*$  of  $O_1$  (formula (5)), 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  $\Theta_2$  of the second level. This process is repeated at each level  $j$  until the corpus likelihood  $P_{MC_n^j}^*(O_j)$  stops increasing:

$$P_{MC_n^j}^*(O_j) \leq P_{MC_n^{j-1}}^*(O_{j-1}), \quad (8)$$

obtaining the optimal level of hierarchy  $\nu$ .

#### 3.1. Maximum Likelihood Estimation of the Model Parameters

As mentioned above,  $O_j$  is the training class corpus at level  $j$ . This corpus is obtained with the most likely segmentation of the class corpus of level  $j - 1$  ( $O_{j-1}$ ). The  $MC_n^\nu$  language model is defined by the set of parameters  $\Theta_j$  ( $j \in \{1 \dots \nu\}$ ) consisting of the probability of each sequence  $s_j(i)$  in a dictionary ( $D_{S_j} =$



$\{s_j(i)\}_{i=1}^m$ . Each sequence  $s_j(i)$  can contain up to  $n$  classes from  $O_j$ .

$$\Theta_j = \{p(s_j(i))\}_{i=1}^m \text{ where } \sum_{i=1}^m p(s_j(i)) = 1 \quad (9)$$

The re-estimation formula of the set of parameters  $\Theta_j$  can be obtained by the Maximum Likelihood (ML) estimation from incomplete data [9], where the observed data are those contained in  $O_j$ , and the unknown data is the underlying segmentation  $\mathcal{S}_j$ . Thus, iterative ML estimates of  $\Theta_j$  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  $k+1$ :

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

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  $\{s_j(i)\}_{i=1}^m$ , can be derived by maximizing the auxiliary function  $Q(k, k+1)$ .

Let  $c(s_j(i), \mathcal{S}_j)$  denote the number of occurrences of the sequence  $s_j(i)$  in a segmentation  $\mathcal{S}_j$  at level  $j$ . We rewrite the joint likelihood given in (3) so as to group together the probabilities of all identical sequences:

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

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:

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

where  $c(\mathcal{S}_j) = \sum_{i=1}^m c(s_j(i), \mathcal{S}_j)$  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 can be used to reestimate formula (13). The set of parameters  $\Theta_j$  can be initialized with the relative frequencies of all co-occurrences of symbols up to length  $n$  in the training corpus. Then  $\Theta_j$  is iteratively re-estimated until the training set likelihood does not increase significantly, or until a fixed number of iterations is reached. Some pruning techniques may be advantageously applied in practice to the dictionary of sequences, in order to avoid over-learning. 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.

## 4. EVALUATION

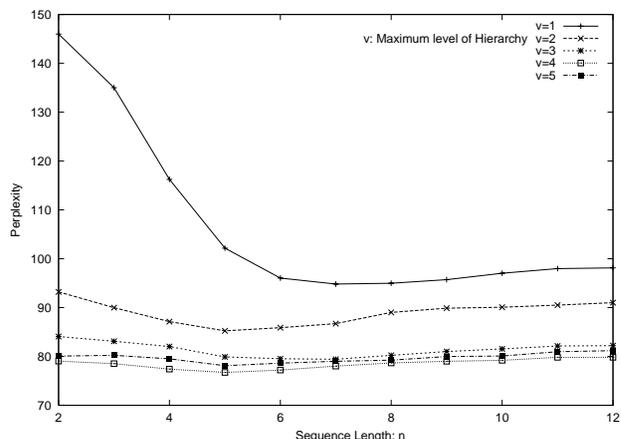
Models have been built for French on a corpus of 43 million words representing two years of "Le Monde" newspaper. The class set used to evaluate the models contains 233 syntactic classes, including punctuation [8]. The vocabulary used contains 20000 words provided by the AUPELF-UREF evaluation campaign (similar to the "Wall Street journal" DARPA test).

### 4.1. Perplexity Results

It is interesting to look at the test perplexity values obtained by the  $MC_n^\nu$  approach at different levels of the hierarchy. This allows us to compare the performance yielded by this approach with a  $n$ -multiclass model and also with a classical interpolated class-grams model (biclass and triclass).

For  $MC_n^\nu$  language model, all co-occurrences symbols are used to get initial estimates of the sequence probabilities. However, to avoid overlearning, we found it 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. In our experiments, the best value of  $C_0$  is equal to 8 ( $C_0 = 8$ ). Then, 10 training iterations are performed in this experiment at each level of the hierarchy (§3.1). Sequence probabilities falling under a threshold  $p_0$  are set to 0, except those of length 1 which are assigned a minimum probability  $p_0$ . The fixed probability was set to  $p_0 \approx 5 \times 10^{-6}$ , i.e. 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 sum to 1. Since all class sequences of length 1 have a minimum probability of  $p_0$ , the likelihood of any string of classes can be computed.

Figure 2 shows the test perplexity obtained by the  $MC_n^\nu$  model for different values of  $n$  and  $\nu$ , as well as the  $n$ -multiclass ( $MC_n^1$ ) test perplexity for different values of  $n$ .



**Fig. 2.** Test perplexity of the  $MC_n^\nu$  model for different values of  $n$  and  $\nu$ , where  $n$  the sequence length and  $\nu$  denotes the maximum level of the hierarchy.

Experiments show that the performance is improving with the increasing of the number of levels in the hierarchy until a value of  $\nu$  equal to 4 with a maximum length of a sequence equal to 5 ( $n = 5$ ). The test perplexity decreases from 145.31 for  $MC_2^1$  to 74.78 for  $MC_5^4$ . The best performance of the  $n$ -multiclass model is obtained with a value of  $n$  equal to 7 ( $PP = 93.52$ ).



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

$n$	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  $n$ -multiclass and the  $MC_n^4$  model with different sequence lengths  $n$ .

The interpolated class  $n$ -grams 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  $n$ -multiclass ( $MC_n^1$ ),  $MC_n^\nu$  and interpolated class  $n$ -grams indicates that  $MC_n^\nu$  outperforms by 85% the performance of a biclass, by 17% the performance of a triclass and by 25% the performance of a  $n$ -multiclass model.

#### 4.2. Contribution of $MC_n^\nu$ to the improvement of MAUD

An evaluation was also done with MAUD [10], our continuous dictation system using a stochastic language model. The basic version of MAUD operates in 3 steps: gender identification, word lattice generation by means of a Viterbi block algorithm with a bigram model, and 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 final result.

To evaluate the performance achieved by the introduction of our model, we used the  $MC_n^\nu$  model to rescore the N-best utterance hypotheses produced by the third step of MAUD: the best hypothesis after rescoring becomes the system result. We also built another version which use the  $n$ -multiclass, instead of the  $MC_n^\nu$  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.

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

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

**Table 2.** Performances of different versions of MAUD system.

Results show that the version using the  $MC_n^\nu$  ( $MC_n^\nu$  **V**) improves the accuracy by 2% compared to the one using the  $n$ -multiclass ( $MC_n^1$  **V**) and by 5% compared the basic one (**BV**).

## 5. CONCLUSION AND PERSPECTIVES

We have described in this paper a new language model which learns statistically hierarchical patterns of word phrases in spoken lan-

guage utterances. This new model is used to rescore the N-best utterance hypotheses which is outputted by a speech recognizer. The hierarchical approach models a sentence as a stream of dependent word sequences according to their syntactic classes.

Experiments show that  $MC_n^\nu$  could be a competitive alternative to the  $n$ -multiclass and interpolated class  $n$ -grams models. The  $MC_n^\nu$  language model outperforms in terms of perplexity the interpolated biclass model by 85%, the triclass model by 17% and the  $n$ -multiclass approach ( $MC_n^1$ ) by 25%. It also outperforms the accuracy of our dictation system MAUD. The MAUD version using the  $MC_n^\nu$  model achieves an improvement of the accuracy by 2% in comparison to the  $n$ -multiclass version and by 5% in comparison to the basic version.

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

## 6. REFERENCES

- [1] S. Deligne and F. Bimbot, "Language modeling by variable length sequences: Theoretical formulation and evaluation of multigrams," in *ICASSP95*, Detroit, Michigan USA, 1995, pp. 169–172.
- [2] S. Deligne, "Statistical language modeling with a class based  $n$ -multigram model," in *ICSLP2000*, Beijing, China, 2000, pp. 119–122.
- [3] S. Deligne, F. Yvon, and F. Bimbot, "Introducing statistical dependencies and structural constraints in variable-length sequence models," *Lecture Notes in Artificial Intelligence 1147*, pp. 156–167, 1996.
- [4] I. Zitouni, K. Smaïli, J.P. Haton, S. Deligne, and F. Bimbot, "A comparative study between polyclass and multiclass models," in *ICSLP98*, Sydney, Australia, Novembre 1998.
- [5] J. Hu, W. Turin, and M.K. Brown, "Language modeling using stochastic automata with variable length contexts," *Computer Speech and Language*, vol. 11, no. 1, pp. 1–16, 1997.
- [6] P.J. Jang and A. Hauptmann, "Hierarchical cluster language modeling with statistical rule extraction for rescoring  $n$ -best hypotheses during speech decoding," in *ICSLP98*, Sydney, Australia, 1998.
- [7] K. Tanigaki, H. Yamamoto, and Y. Sagisaka, "A hierarchical language model incorporating class-dependent word models for oov words recognition," in *ICSLP2000*, Beijing, China, 2000, pp. 123–126.
- [8] K. Smaïli, F. Charpillet, and J.P. Haton, "A new algorithm for automatic word classification based on an improved simulating annealing technique," in *5th International Conference in Cognitive Science and Natural Language Processing*, Dublin, Ireland, 1996.
- [9] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum-likelihood from incomplete data via the em algorithm," *Journal of Royal Statistic Society, Ser. B (methodological)*, , no. 39, pp. 1–38, 1977.
- [10] D. Fohr, J.P. Haton, J.F. Mari, K. Smaïli, and I. Zitouni, "Towards an oral interface for data entry: The maud system," in *3rd ERCIM Workshop on "User Interface for All"*, Obernai, France, November 1997.