Variable-Length Class Sequences Based on a Hierarchical Approach: MCnv
Imed Zitouni, Kamel Smaïli, Jean-Paul Haton

To cite this version:

HAL Id: inria-00099045
https://inria.hal.science/inria-00099045
Submitted on 26 Sep 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
VARIABLE-LENGTH CLASS SEQUENCES BASED ON A HIERARCHICAL APPROACH: MC$^{n/\nu}$

Imed ZITOUNI, Kamel SMAIL, 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$^{n/\nu}$, 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 capability to take into account 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$^{n/\nu}$ 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. Results 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

recognition. Most current language models are based on n-gram modeling to solve the problem of data sparseness, 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.

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$^{n/\nu}$ 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
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 \( W(w_1, w_2, \ldots, w_n) \) how to determine the syntactic categories \( C(c_1, c_2, \ldots, c_n) \) that maximize:

\[
P(c_1 \ldots c_n / w_1 \ldots w_n) = \frac{P(c_1 \ldots c_n) P(w_1 \ldots w_n / c_1 \ldots c_n)}{P(w_1 \ldots w_n)}. \tag{1}
\]

As we are interested in finding \( c_1c_2 \ldots c_n \), 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:

\[
P(c_1 \ldots c_n/w_1 \ldots w_n) = P(c_1)P(w_1/c_1)P(c_1/c_2)P(w_2/c_2) \times \prod_{i=3}^{n} P(c_i/c_{i-2}c_{i-1})P(w_i/c_i). \tag{2}
\]

In order to estimate the probabilities \( P(c_1/c_{i-2}c_{i-1}) \) 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 \( P(c_1/c_{i-2}c_{i-1}) \) can be expressed as a relative frequency

\[
P(c_1/c_{i-2}c_{i-1}) = \frac{n(c_1 \ldots c_{i-2}c_{i-1})}{n(c_{i-2}c_{i-1})} \tag{3}
\]

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:

\[
P(w_i/c_i) = \frac{n(w_i/c_i)}{n(c_i)} \tag{4}
\]

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\(_n\) 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 \( \nu \) as MC\(_{n,\nu}\). Using this notation, the traditional multiclass can be written as MC\(_{n,1}\).

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). After, at each level \( j \) (\( j \in \{1 \ldots \nu\} \)) 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 \( \nu \). 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). In figure 1 we present an example of applying the MC\(_{n,2}\) 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 \( n \)
5 FORMULATION OF THE MC\textsuperscript{\nu} MODEL

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

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

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

$$P(C) = P_{MC\textsuperscript{\nu}}(\Omega_\nu) \quad (6)$$

where $\Omega_\nu$ is a class phrase corresponding to the most likely segmentation $S_{MC\textsuperscript{\nu}-1}^*$ of the class phrase of level $\nu - 1$ ($\S$formula 9). $\Omega_{\nu-1}$. This class phrase $\Omega_{\nu-1}$ corresponds also to the most likely segmentation of $\Omega_\nu$ (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 $C$. If $\nu = 1$ the $MC\textsuperscript{\nu}$ model is similar to the basic n-multiclass model.

Let $S_j$ be a possible segmentation of the class phrase $\Omega_j$ building $(s_j(1), s_j(2), \ldots, s_j(q_j))$, where $s_j(i)$ corresponds to a class sequence in $\Omega_i$. The likelihood $P_{MC\textsuperscript{\nu}}(\Omega_j, S_j)$ of the class phrase $\Omega_j$ associated with segmentation $S_j$ 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 $n$:

$$P_{MC\textsuperscript{\nu}}(\Omega_j, S_j) = \prod_{t=1}^{q_j} p(s_j(t)) \quad (7)$$

Denoting as $\{S_j\}$ the set of all possible segmentation of $\Omega_j$ into class sequences, the likelihood of $\Omega_j$ is:

$$P^*_j(\Omega_j) = \max_{S_j \in \{S_j\}} P_{MC\textsuperscript{\nu}}(\Omega_j, S_j) \quad (8)$$

where the most likely segmentation $S^*_j$ of $\Omega_j$ is:

$$L^*_j = \arg P^*_j(\Omega_j) = \Omega_{j+1}. \quad (9)$$

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 $C = abcd$ ($P(C)$) is computed in an increasing way. We denote sequence borders with brackets. For $j = 1$ ($\text{\Omega}_1 = \text{abed}$):

$$P^*_1(\text{\Omega}_1) = \max \left\{ \begin{array}{l} p([a]) p([bcd]) \ p([a]) p([bcd]) \ p([a]) p([bcd]) \ p([a]) p([bcd]) \ p([a]) p([bcd]) \ p([a]) p([bcd]) \ p([a]) p([bcd]) \ p([a]) p([bcd]) \end{array} \right\}$$

Assume that $P_{MC\textsuperscript{\nu}}^i(\Omega_i) = P^{i+1}(\text{[a]} | p([bc]) | p([d]))$ and let $X$ be the new tag of the class sequence $\text{[bc]}$ ($X \equiv [bc]$). Let $\Omega_2 = aXd$ and

$$P(C) = P^*_2(\Omega_2) = \max \left\{ \begin{array}{l} p([a]) p([Xd]) \ p([a]) p([d]) \ p([a]) p([d]) \ p([a]) p([d]) \ p([a]) p([d]) \end{array} \right\}$$

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 $\nu$ 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 $j D_\nu$: $\Theta_j = \{p(s_j(i))\}$, with $\sum_{s_j(i)} p(s_j(i)) = 1$. $D_\nu = \{s_j(i)\}$ denotes a dictionary of class sequences which can be formed by combining 1, 2, . . . up to $n$ classes from the training class corpus of level $j$ $(O_j)$. The most likely segmentation $S^*_j$ 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 $\Theta_{j+1}$. $\Theta_1$ 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 $\Theta_1$. Then, we build the most likely segmentation $S^*_j$ of $O_1$ ($\S$formula 9), obtaining the second level training corpus $O_2$ with a likelihood equal to $P^*_1(\text{O}_1)$. The class corpus $O_2$ is used to extract sequences and to estimate the set of parameters $\Theta_2$ of the second level. We repeat this process at each level $j$ until the corpus likelihood $P^*_j(\text{O}_j)$ stop increasing:

$$P^*_j(\text{O}_j) \leq P^*_j-1(\text{O}_{j-1})$$

obtaining the optimal level of hierarchy $\nu$.

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 $MC\textsuperscript{\nu}$ language model is completely defined by a set of parameters $\Theta_j (j \in \{1 \ldots \nu\})$ consisting of the probability of each sequence $s_j(i)$ in a dictionary $(D_j = \{s_j(i)\} \cap [n])$. Each sequence $s_j(i)$ can contains until $n$ classes from $O_j$.

$$\Theta_j = \{p(s_j(i))\} \cap [n] \text{ where } \sum_{i=1}^{n} p(s_j(i)) = 1$$

The re-estimation formula of the set of parameters $\Theta_j$ 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 $S_j$. 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 $k + 1$:

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

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
Let \( c(s_j(i), S_j) \) denote the number of occurrences of the sequence \( s_j(i) \) in a segmentation \( S_j \) 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:

\[
P^{(k+1)}(O_j, S_j) = \prod_{i=1}^{m} p^{(k+1)}(s_j(i))^{c(s_j(i), S_j)}
\]

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

\[
Q(k, k + 1) = \sum_{i=1}^{m} \sum_{S_j \in \{S_j\}} P^{(k)}(O_j, S_j) c(s_j(i), S_j) \log p^{(k+1)}(s_j(i)).
\]

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]:

\[
p^{(k+1)}(s_j(i)) = \sum_{S_j \in \{S_j\}} c(s_j(i), S_j) \times P^{(k)}(O_j, S_j) \sum_{S_j \in \{S_j\}} c(S_j) \times P^{(k)}(O_j, S_j)
\]

where \( c(S_j) = \sum_{i=1}^{m} c(s_j(i), S_j) \) is the total number of sequences in \( S_j \). 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 \( \Theta_j \) 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 [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\(_n^+\) 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\(_n^+\) model with the class n-gram and classical multiclass (MC\(_n^\)) models.

### 6.1 Data Description

Models have been built on a French corpus (LeM) of 43 million 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\(_n^+\) 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 \( W \) (\( T \) words) tagged by the class corpus \( C \) is computed as follows:

\[
PP(W) = 2^{-\frac{1}{T} \sum_{t=1}^{T} \log P(W_t)}
\]

where \( P(W) \) is the likelihood of the test corpus. In the case of MC\(_n^+\) and multiclass (MC\(_n \)) models, this likelihood value \( P(W) \) is computed according to the formula 5.

For MC\(_n^+\) 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\(_n^+\) model for different values of \( n \) and \( \nu \). One can notice that figure 2 shows also the multiclass (MC\(_n^\)) test perplexity for different values of \( n \).

Experiments show that, progressively the number of hierarchy increases, the performances improve until a value of \( \nu \) 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\(_2^+\) and decrease to 74.78 with a number of hierarchy equal to 4 and with sequence length equal to 5 \( (MC_5^+\)\). 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\(_2^\)) and the MC\(_n^+\) 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\(_n^\)), MC\(_n^+\) and interpolated class n-gram indicates that MC\(_n^+\) outperform by 85% the performance of a biclass, by 17% the performance of a triclass and by 25% the performance of a multiclass model.
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 the 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\textsubscript{n} 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 uses the multiclass, instead of the MC\textsubscript{n} 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 these different rates for the basic version (BV), the version using multiclass (MC\textsubscript{n} V) and the one using MC\textsubscript{n} model (MC\textsubscript{n} V) with the optimal level of hierarchy (\(\nu = 4\)).

Results show that the version using the MC\textsubscript{n} (MC\textsubscript{n} V) improves the accuracy by 2\% compared to the one using the multi-class (MC\textsubscript{n} V) and by 5\% compared to the basic one (BV).

### 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\textsubscript{n} could be a competitive alternative to the multiclass and interpolated class n-gram models. The MC\textsubscript{n} language model outperforms in terms of perplexity the interpolated biclass model by 85\%, the triclass model by 17\% and the multiclass approach (MC\textsubscript{3}) 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\textsubscript{n} 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\textsubscript{n} 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 Frédéric BIMBOT and Sabine DELIGNE for the package of multigram and for the faithful discussions we have about this new approach.

### 8 References


<table>
<thead>
<tr>
<th>(n)</th>
<th>PP(_{MC_1})</th>
<th>PP(_{MC_2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>115.91</td>
<td>75.95</td>
</tr>
<tr>
<td>5</td>
<td>100.02</td>
<td>74.78</td>
</tr>
<tr>
<td>6</td>
<td>94.92</td>
<td>75.17</td>
</tr>
<tr>
<td>7</td>
<td>93.52</td>
<td>75.67</td>
</tr>
<tr>
<td>8</td>
<td>93.81</td>
<td>75.94</td>
</tr>
</tbody>
</table>

Table 1: Test perplexity of the multiclass and the MC\textsubscript{n} model with different sequence lengths \(n\).


