# A COMPARATIVE STUDY BETWEEN POLYCLASS AND MULTICLASS LANGUAGE MODELS

*I. Zitouni*, K. Smaïli*, J.P. Haton*, S. Deligne+⊗, F. Bimbot+×*

*LORIA, BP 239, 54506 Nancy, France

×IRISA-CNRS/INRIA, Campus universitaire de Beaulieu, 35042 Rennes cedex, France

+ENST-Dept Signal, CNRS-URA 820, 75634 Paris cedex 13, France

⊗ATR Interpreting Telecommunication Labs, Kyoto, Japan

E-mail: {zitouni, smaili, jph}@loria.fr, sdeligne@itl.atr.co.jp, bimbot@irisa.fr

## ABSTRACT

In this work, we introduce the concept of Multiclass for language modeling and we compare it to the Polyclass model. The originality of the Multiclass is its capability to parse a string of classes/tags into variable length independent sequences. A few experimental tests were carried out on a class corpus extracted from the French « Le Monde » word corpus labeled automatically. This corpus contains a set of 43 million of words. In our experiments, Multiclass outperform first-order Polyclass but are slightly outperformed by second-order Polyclass.

## 1. INTRODUCTION

Language can be viewed as a stream of words emitted by a source. This language source being subject to syntactic and semantic constraints, words are not independent, and the dependencies are of variable length. One can therefore expect to retrieve, in a corpus, typical variable-length sequences of words. The Multiclass model, presented in this paper, is an application of the Multigram model [1] to sequences of classes for modeling these variable-length dependencies. To deal with the syntactic constraints in a language, we label the stream of words with 233 classes (a word can belong to several classes) extracted from the eight elementary grammatical classes of the French language. This paper presents a comparison of the Multiclass language model with the n-class model which is a kind of a generalization of the most used language model in the speech recognition community (the n-gram). The n-class model used in this paper is an interpolated n-class named Polyclass. This language model is based on the same principles as the n-gram language model with this difference that classes are used instead of words. In the following we first discuss the necessity and the manner of tagging a corpus of text (Section 2). Second, we introduce the concept of the Polyclass language model used for the comparison (Section 3). Third, we give a theoretical background of the Multiclass language model (Section 4). Then, we report an evaluation of the Multiclass model and a comparison with the Polyclass model (Section 5). Finally, we give a conclusion and some perspectives.

## 2. THE NECESSITY OF TAGGING

The concept of class is very important in the two methods presented below. We explain in this section how we proceeded to tag our corpus using a set of syntactic tags. The problem at hand is the following: given a sentence $W(w_1w_2...w_n)$, how to label the words of W with the syntactic categories $C(c_1c_2...c_n)$ in a way which maximizes:

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

As we are interested in finding $c_1c_2...c_n$, the denominator will not affect the computation. By making some independence assumptions, formula (1) can be expressed as:

$$P(c_1c_2\cdots c_n / w_1w_2\cdots w_n)=\prod_{i=1}^{n}P(c_i / c_{i-2}c_{i-1})P(w_i / c_i) \quad (2)$$

In order to estimate the probability $P(c_i / c_{i-2}c_{i-1})$, 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 eight elementary grammatical classes of French, we built up *233* classes including punctuation. Labeling the words of the vocabulary with these *233* classes resulted in a dictionary of *230 000* entries, each of which consists of a word and its syntactic 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 co-occurrences of the syntactic tags specified by $x$ in a training text. The first step consists in 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. The errors resulting from this automatic tagging were hand-corrected, and the updated label statistics were used to automatically tag another, larger, set of *43*

million words, consisting of *2* years (1987-1988) of *Le Monde* (LeM) newspaper. Tagging a corpus means to find the most likely sequence of classes for a sequence of words. In our approach we used a modified Viterbi algorithm [2].

## 3. POLYCLASS MODEL

Like in Multiclass, we use a corpus of tags/classes obtained by labeling a text corpus. Each word of the corpus is a syntactic class. A Polyclass model is a language model which takes into account only classes. The formalism of Polyclass language model can be expressed as :

$$P(C_1C_2....C_n) = \prod_{i=1}^{n} p(C_i / h_i) \qquad (4)$$

where $h_i$, the history of $C_i$, is a fixed-length sequence of classes. We call the length of the history the order of the Polyclass model. In the following, we use only second and first order Polyclass. Even though the set of distinct syntactic tags is much smaller than the size of the vocabulary (*233* tags versus thousands of words), most combinations of class labels occur only a few times in any. In our corpus LeM, *34%* of the observed 3-class and more than *15%* of the observed 2-class occur only once, and, *34%* of the observed 2-class and *62%* of the observed 3-class occur *5* times or less. The errors resulting from the automatic tagging tend to enhance the inherent sparseness of the data. In order to get reliable estimates, the probabilities $P(C_i/h_i)$ have thus to be smoothed [3]. For this purpose, we used an interpolation scheme, where the relative counts of the 3-class ($h_i$ consists of the *2* class labels preceding $C_i$) are linearly interpolated with the relative counts of the 2, 1 and 0-class:

$$\alpha\, p(c_i / c_{i-2}c_{i-1}) + \beta\, p(c_i / c_{i-1}) + \gamma\, p(c_i) + \delta \qquad (5)$$

where $\alpha + \beta + \gamma + \delta = 1$.

The interpolation weights $\alpha$, $\beta$, $\gamma$ and $\delta$ were estimated by maximizing the likelihood of a development corpus. For this purpose, we used the algorithm proposed by Jelinek & al., who showed in [3] that the ML estimation of the interpolation weights could be assimilated to the ML estimation of the transition probabilities of an HMM, thus allowing to use the forward-backward algorithm classically used in the HMM framework.

## 4. MULTICLASS MODEL

In the Multiclass approach, derived from the Multigram framework, string of classes are assumed to result from the concatenation of variable-length sequences of classes, of maximum length *n* class labels. The likelihood of a string of classes is computed by summing the likelihood values of all possible segmentations of the string into sequences of classes. By denoting by *L* a segmentation of a string *C* of classes:

$$P(C) = \sum_{L \in \{L\}} P(C, L) \qquad (6)$$

The decision-oriented version of the model parses *C* according to the most likely segmentation, thus yielding the approximation:

$$P^*(C) = \max_{L \in \{L\}} P(C, L) \qquad (7)$$

The likelihood computation for any particular segmentation into sequences depends on the model assumed to describe the dependencies between the sequences. Assuming that the sequences of classes are independent, it comes:

$$P(C, L) = \prod_{t=1}^{t=q} p(s(t)) \qquad (8)$$

where *s(t)* denotes the $t^{th}$ sequence of classes in the segmentation *L* of *C*. The model is thus fully specified by the set of probabilities, $\{p(s_i)\}_i$, of all the sequences $s_i$ which can be formed by combining *1*, *2*, … or *n* class labels.

Maximum likelihood estimates of these probabilities can be computed by formulating the estimation problem as an ML estimation from incomplete data [5], where the observed data is the string of symbols *C*, and the unknown data is the underlying segmentation L. Denoting by $nb(s_i, L)$ the number of occurrences of the sequence $s_i$ in a segmentation *L* of the corpus, at iteration *k+1* the probability of the sequence $s_i$ is obtained [4]:

$$p^{(k+1)}(s_i) = \frac{\sum_{L \in \{L\}} nb(s_i, L) \times P^{(k)}(C, L)}{\sum_{L \in \{L\}} nb(L) \times P^{(k)}(C, L)} \qquad (9)$$

where $\mathrm{nb(L)} = \sum_{i=1}^{m} \mathrm{nb(s_i, L)}$ is the total number of sequences in *L*. Equation (9) shows that the estimate for $p(s_i)$ is merely a weighted average of the number of occurrences of sequence $s_i$ within each segmentation. Since each iteration improves the model in the sense of increasing the likelihood $P^{(k)}(C)$, it eventually converges to a critical point (possibly a local maximum).

The reestimation (9) can be implemented by means of a forward-backward algorithm [4]. The set of initial probabilities can be initialized with the relative frequencies of all co-occurrences of symbols up to length *n* in the training corpus. Then the probabilities are iteratively reestimated until the training set likelihood does not increase significantly, or with a fixed number of iterations. In practice, some pruning technique may be advantageously applied to the dictionary of sequences, in order to avoid over-learning. A straightforward way to proceed consists in simply discarding, at each iteration, the most unlikely sequences, i.e. those with a probability value falling under a prespecified threshold.

# 5. EVALUATION

In this section, we present a comparative evaluation of the Polyclass and of the Multiclass models, based on experiments on the LeM corpus. For each experiment, we used a vocabulary of *233* classes including punctuation extracted from the eight elementary grammatical classes of the French language [6]. These classes are divided into two groups: the open 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, ...). Each punctuation symbol is in a single class. The performance of the Multiclass and the Polyclass are evaluated in terms of class perplexity [7]:

$$PP = 2^{-\frac{1}{T}\log_2 P(C)}$$

where $T$ is the number of syntactic tags in a test set $C$. In the Multiclass case, $P(C)$ is computed from equation (6).

The first experiment concerns the Polyclass language model. In this experiment, the Polyclass relative counts are computed on a training set of 40 millions of classes, and the interpolation weights ($\alpha$, $\beta$, $\gamma$, $\delta$) on an additional development set of 1,8 millions of classes. Test perplexity values are computed on a distinct test set of about 1,6 millions of classes. The corpus of development and test do not appear in the training corpus. Table 1 shows the results obtained for a first and second order Polyclass model and gives the values of the interpolation weights.

| Order | $\alpha$ | $\beta$ | $\gamma$ | $\delta$ | Nb | PP |
|---|---|---|---|---|---|---|
| 1 | 0 | $9{,}99\times10^{-1}$ | $6{,}57\times10^{-5}$ | 0 | 17 500 | 13,59 |
| 2 | 0,997 | $2{,}04\times10^{-3}$ | $6{,}57\times10^{-5}$ | 0 | 265 000 | 11,03 |

**Tab1 : This table shows for each Polyclass model with an order of 1 and 2 the values of the interpolation weights, the number of parameters in the model Nb, and the Polyclass perplexity PP on a test corpus of 1,6 millions of classes.**

In a second series of experiments, we compare the Polyclass and The Multiclass models on only one month (Jan87) of LeM corpus, which we split into a training corpus, a development corpus and a test corpus. We use a training corpus of *55000* class sentences (more than *1,7* million of classes), a test corpus of *5000* class sentences (more than *0,15* million classes) and a development corpus of *3000* class sentences (more than *0,1* million classes). In the Polyclass model, the development corpus is used to evaluate the parameters $\alpha$, $\beta$, $\gamma$, $\delta$, and in the Multiclass model we use this corpus to optimize the maximum number (n) of classes in a Multiclass sequence and the number of occurrences ($C_0$) above which a sequence of words is included in the initial inventory of sequences. The corpora of development and test do not appear in the training corpus.

For the Multiclass 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 $C_0$. Then, 10 training iterations are performed in this experiment with different values of $n$ and $C_0$. 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 $P_0 \approx 5\times10^{-6}$ 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].

| n | | $C_0$=0 | $C_0$=1 | $C_0$=2 | $C_0$=5 | $C_0$=10 |
|---|---|---|---|---|---|---|
| 3 | $PP_{Tr}$ | 14,08 | 14,19 | 14,25 | 14,35 | 14,47 |
| | $PP_{D}$ | 14,98 | 14,85 | 14,83 | 14,85 | 14,91 |
| | $PP_{Tst}$ | 14,79 | 14,63 | 14,61 | 14,64 | 14,70 |
| | Nb | 25034 | 20663 | 17941 | 13708 | 10490 |
| 5 | $PP_{Tr}$ | 9,65 | 10,87 | 11,27 | 11,77 | 12,15 |
| | $PP_{D}$ | 18,51 | 13,20 | 12,86 | 12,77 | 12,89 |
| | $PP_{Tst}$ | 18,31 | 12,95 | 12,58 | 12,48 | 12,61 |
| | Nb | 125876 | 98120 | 78223 | 50568 | 33696 |
| 8 | $PP_{Tr}$ | 5,02 | 8,96 | 10,03 | 11,04 | 11,63 |
| | $PP_{D}$ | 21,59 | 13,34 | 12,52 | 12,32 | 12,50 |
| | $PP_{Tst}$ | 21,57 | 13,25 | 12,35 | 12,03 | 12,19 |
| | Nb | 188994 | 156376 | 117525 | 67943 | 41492 |

**Tab 2 : This table shows the number of learning parameters (Nb), the perplexity on the training corpus ($PP_{Tr}$), the perplexity on the development corpus ($PP_D$) and the perplexity on the test corpus ($PP_{Tst}$) for different number of n and $C_0$. *n* is the maximum number of words in a Multiclass sequence and $C_0$ is the number of occurrences above which a sequence of words is included in the initial inventory of sequences.**

The experiments of Table 2 show that the minimum perplexity is for $n \geq 8$ and $C_0 \approx 5$. Other experiments with $n \in \{7,8,9,10\}$ and $C_0 \in \{4,5,6,7\}$ are reported in Table 3.

The experiments (Table 3) show that the minimum perplexity (*12,00*) on the test corpus is obtained with $n=10$ and $C_0=4$. The comparison of perplexity of both Multiclass and Polyclass indicates that from $n=5$ and $C_0 \geq 1$, the Multiclass is better than the first order Polyclass (*13,46*) but gives less good results than second order Polyclass (*11,43*). It is important to note that the number of units is in the same order of magnitude for optimal Multiclass and the second order Polyclass ($\approx$*70000* for Multiclass vs *80000* for second order Polyclass).

| n | | $C_0$=4 | $C_0$=5 | $C_0$=6 | $C_0$=7 |
|---|---|---|---|---|---|
| 7 | $PP_D$ | 12,35 | 12,37 | 12,39 | 12,44 |
| | $PP_{Tst}$ | 12,09 | 12,07 | 12,08 | 12,11 |
| | Nb | 77359 | 66919 | 59378 | 53207 |
| 8 | $PP_D$ | 12,28 | 12,32 | 12,36 | 12,41 |
| | $PP_{Tst}$ | 12,04 | 12,03 | 12,04 | 12,07 |
| | Nb | 78792 | 67943 | 60055 | 53801 |
| 9 | $PP_D$ | 12,25 | 12,29 | 12,33 | 12,38 |
| | $PP_{Tst}$ | 12,02 | 12,00 | 12,02 | 12,06 |
| | Nb | 78661 | 67820 | 59957 | 53600 |
| 10 | $PP_D$ | 12,24 | 12,28 | 12,32 | 12,37 |
| | $PP_{Tst}$ | 12,00 | 12,00 | 12,01 | 12,05 |
| | Nb | 78130 | 67355 | 59552 | 53239 |

**Tab 3 : This table shows the number of learning parameters (Nb), the perplexity on the development corpus ($PP_D$) and the perplexity on the test corpus ($PP_{Tst}$) for n∈{7,8,9,10} and $C_0$ ∈{4,5,6,7}. *n* is the maximum number of words in a Multiclass sequence and $C_0$ is the number of occurrences above which a sequence of words is included in the initial inventory of sequences.**

| Order | α | β | γ | δ | Nb | PP |
|---|---|---|---|---|---|---|
| 1 | 0 | $9,98x10^{-1}$ | $1,29x10^{-3}$ | 0 | 9 100 | 13,46 |
| 2 | 0,981 | $1,73x10^{-2}$ | $1,29x10^{-3}$ | 0 | 80 000 | 11,43 |

**Tab4 : The table shows for each Polyclass model with an order of 1 and 2 the values of the necessary parameters, the number of learning parameters Nb and the class perplexity PP on a corpus of 55000 classes.**

Table 4 shows the results obtained for a Polyclass model using respectively a length history of 1 (order 1) and 2 (order 2).

# 6. CONCLUSION AND PERSPECTIVES

The experiments reported in this paper show that the Multiclass approach is a competitive alternative to the Polyclass (n-class) language model. On our task, the Multiclass language model outperforms in terms of perplexity the first order Polyclass model (2-class interpolated with the 1-class and 0-class), but we note that the Multiclass model gives slightly less good results than the second order Polyclass. In order to improve the Multiclass model, we will study methods for interpolating the sequence probabilities. Another direction consists in assuming dependencies between the sequences of classes as is proposed in [8][9][10]. It also seems interesting to investigate the application of the Multiclass approach to other issues. Indeed, this approach might be advantageously used to filter the lattice or the N-best list of sequences output by a speech recognizer, for instance by supplying information on semantic equivalence between sequences of words. More generally, it may find applications in the area of language understanding, such as concept tagging based on the labels of phrase classes.

# 7. REFERENCES


1. F. Bimbot, R. Pieraccini, E. Levin and B. Atal « Variable-Length Sequence Modeling: Multigrams ». IEEE Signal Processing Letters, N° 6, Vol. 2 , Jun 1995.
2. K. Smaïli, I. Zitouni, F. Charpillet, J.-P. Haton « An Hybrid Language Model for a Continuous Dictation Prototype », 5th European Conference on Speech Communication and Technology , PP 2723-2726, Vol 5, Rhodes 1997.
3. F. Jelinek, R. Mercer and S. Roukos « Principles of Lexical Language Modeling for Speech Recognition », in Avances in Signal Processing, Furui S. editor, New-York, Marcel Dekker, PP 651-699, 1992.
4. S. Deligne and F. Bimbot. "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams". ICASSP95, PP 169-172, 1995.
5. A.P. Dempster, N.M. Laird, and D.B. Rubin. « Maximum-likelihood from incomplete data via the EM algorithm ». Journal of the Royal Statistical Society, 39(1):1-38, 1977.
6. K. Smaïli, F. Charpillet and JP. Haton "A new Algorithm for Word Classification based on an Improved Simulated Annealing Technique". 5th International Conference on the Cognitive Science of Natural Language Processing, Dublin, 1996.
7. F. Jelinek (1990). Self-organized language modeling for speech recognition, in Reading in Speech Recognition, pp.450-506. Ed. A.Waibel and K.F.Lee editor. Morgan Kaufmann Publishers Inc.,San Mateo, California, 1990.
8. I. Zitouni, K. Smaïli and JP. Haton. Variable-Length Class Sequences Based on a Hierarchical Approach: $MC^v_n$. will appear in International workshop SPEECH and COMPUTER. St Petersburg (1998).
9. S. Deligne and Y. Sagisaka. Learning a Syntagmatic and Paradigmatic Structure from Language Data with a bi-multigram Model. Proceedings of COLING-ACL 98, August 1998.
10. 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, PP 156-167, Springer 1996.