An Hybrid Language Model For a Continuous Dictation Prototype

K. Smaïli, I. Zitouni, F. Charpillet and J-P. Haton
CRIN-CNRS/INRIA Lorraine
BP 239 54506 Vandoeuvre Lès-Nancy France
E-mail : {smaili, zitouni, charp, jph}@loria.fr
Tel : (33) 03-83-59-20-83 Fax : (33) 03-27-83-29

Abstract

This paper describes the combination of a stochastic language model and a formal grammar modelled such as a unification grammar. The stochastic model is trained over 42 million words extracted from *Le monde* newspaper. The stochastic model is based on smoothed 3-gram and 3-class. The 3-class model is represented by a Markov chain made up of four states. Several experiments have been done to state which values are the best for specific training and test corpus. Experiments indicate that the unification grammar reduce strongly the number of hypothesis (sentences) produced by the stochastic model.

1. Introduction

The oral entry of texts (dictation machine) remains an important challenge for the research community in speech recognition. Our research group is working in this area for over 15 years, in particular in the framework of the MAUD project. In the prototype, we have recently developed, a second-order Hidden Markov Model (HMM2) [1] is used to recognise words. The purpose of this paper is to present a language model which is based on a combination of both stochastic and a formal grammar. The stochastic model is based on smoothed 3-gram and 3-class trained over a corpus of 42 million words. In the next sections, we describe MAUD's functioning and then we detail the different aspects of the language model we propose and we finish by giving some results which have been obtained in the framework of the AUPELF project.

2. Description of MAUD

MAUD is a 20K words continuous dictation system using a stochastic language model. The acoustic model is trained on BREF database. MAUD is fundamentally based on a stochastic approach and proceeds in 4 steps : Gender identification, building a word lattice, N-best sentences building and sentence filtering

2.1 Gender Identification

The signal is parametrised with 12 MFCC coefficients, with their first and second derivatives. Each frame is computed every 8ms. Two recognition systems are used in parallel : the first one uses 35 context independent phonetic models constructed for male speakers, while the second one uses a similar model dedicated for female speakers. The best likelihood algorithm determines the speaker gender. In this step, we use a very narrow beam search in the recogniser, in order to accelerate the identification process.

2.2 Building a word lattice

The goal of this step is to build a word lattice from the speech signal. For that, a context dependent acoustic models are used according to the results of the first step. Each phoneme in context (diphone) is modelled by a second order Markov model with 3 states (HMM2). Each word in the lexicon is represented by the concatenation of the HMM2 diphones which compose it. To obtain the word lattice, our system uses a slightly modified block-Viterbi algorithm [2] which takes into account the usual phonological alterations (deletion, liaisons,...) of spoken French language and a bigram langage model.

2.3 N-best sentences building

This step builds the N-best sentences using the word lattice obtained at the previous step and a Trigram language model. A beam search is used accounting for both acoustic scores and alignment calculated during the previous step (no acoustic recalculation is required). The result is a list of ordered sentences in accordance with combined score of the acoustic and language models.

2.4 Sentence filtering

Sentence filtering is carried out using a probabilistic model based on 2-class and 3-class improved by grammatical rules which reduce the ambiguities inherent to a positional classic model. These grammatical rules are based on the unification grammar formalism. The aim of this grammar is to take into account phenomena such as agreement in gender and number. The sentences outputted in the previous step are syntactically labelled and are filtered in order to keep the *N* best sentences according to the formula (7) which will be detailed further in the paper. The N best sentences kept are examined by the unification grammar in order to eliminate the sentences which do not respect the agreement grammatical rules.

3. Stochastic Language Model

Despite an explicit formal grammar for natural language is more expressive, stochastic n-gram language models are still preferred for building operational large vocabulary speech recognition systems because they can be trained on large corpora.

Consider the problem of recovering a sentence W from an acoustic signal A. This problem is usually solved by maximising formula (1).

$$P(W / A) = \frac{P(A / W) P(W)}{P(A)} \qquad (1)$$

Where $P(W / A)$ denotes the probability that the sentence $W(w_1 w_2 ... w_n)$ was uttered knowing A, $P(A / W)$ is the probability that W corresponds to the signal A and $P(A)$ is the average probability that A will be observed. The most important task for the language model is to compute precisely the term $P(W)$. The purpose of language model is to compute $P(w_1 w_2 ... w_n)$. This probability is often approximated by :

$$P(w_1 w_2 \cdots w_n) = \prod_{i=1}^{n} P(w_i / h_i) \qquad (2)$$

Where h_i is a more or less long history. The history is generally made up of a sequence of words, but we argue that it can be generalised by using semantic or syntactic attributes. We use both the classical model with smoothed trigram to estimate the formula (1) during the word lattice construction phase and a higher model which take into account syntactic contexts for filtering.

For that, it is important to formulate the problem of learning a syntactic language model: given a sentence $W(w_1 w_2 ... w_n)$ how to determine the syntactic categories $C(c_1 c_2 ... c_n)$ that maximises

$$P(c_1 \cdots c_n / w_1 \cdots w_n) = \frac{P(c_1 \cdots c_n) P(w_1 \cdots w_n / c_1 \cdots c_n)}{P(w_1 \cdots w_n)} \qquad (3)$$

As we are interested in finding $c_1 c_2 ... c_n$ the common denominator will not affect the computation. By making some independence assumptions, the formula 3 [3] can be expressed as

$$P(c_1 c_2 \cdots c_n / w_1 w_2 \cdots w_n) = \prod_{i=1}^{n} P(c_i / c_{i-2} c_{i-1}) P(w_i / c_i) \qquad (4)$$

In the next section, we will discuss the way to compute $P(c_i / c_{i-2} c_{i-1})$ and the necessity to give to each word the syntactic classes it belongs to.

4. The necessity of taging

Given the formula (4), we can easily understand that text or speech has to be labelled syntactically. 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 need a syntactic field for each entry. This involves that some words have to be duplicate if they appear in more than one class. From the eighth elementary grammatical classes of French, we build up about 230 classes including punctuation [3]. 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 opened 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 probability $P(c_i / c_{i-2} c_{i-1})$ can be expressed as a relative frequency

$$P(c_i / c_{i-2} c_{i-1}) = \frac{n(c_{i-2} c_{i-1} c_i)}{n(c_{i-2} c_{i-1})} \qquad (5)$$

Where $n(x)$ counts the number of times that the syntactic structure x occurs in a training text. So one of the first steps to do is to collect the counts of 3-class (a sequence of 3 classes) and 2-class (a sequence of 2 classes). For that, we labelled 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* newspaper. This tagging has been checked by hand and the automatic labelling errors have been corrected. After, we labelled automatically a corpus of 42 million words which represent 2 years (1987-1988) of *Le Monde* (LeM) newspaper. To tag a corpus mean to find the most likely sequence of classes for a sequence of words. One way to do that is to use the Viterbi algorithm. In our approach, we developed an algorithm based on the dynamic time warping. We build up a probabilistic network where each node is associated to a 2-class and the transition from one state to another produces a 3-class. To each word of a test corpus we assign all the classes to which it belongs to and not all the classes as in Viterbi. To make this operation possible, we use a dictionary of 230 000 words. When a word has to be labelled and it does not appear in the dictionary, we assign to this word all the opened classes defined in our classification. Assume $G = (X, \psi)$ is the network associated to a sequence of words to label ; X is the set of its states, $\psi(x)$ the set of the successors of the state x and $P(x,y)$ the probability to reach the state y from x with $y \in \psi(x)$. In this case the formula, we used to calculate the cost of the best labelling is given by

$$F_i(j) = \max_k F_{i-1}(k) P(k, j) \quad (6)$$

with $k \in \psi^{-1}(j)$, $\psi^{-1}(x)$ is the set of predecessors of the state x of G and $F_i(j)$ is the probability to reach the state j at step i.

5. Getting reliable statistics

One of the problem of the stochastic language model is the unseen events. In formula 5, if the 2-class event $c_{i-2}c_{i-1}$ never appears in the training corpus, formula 5 will not be computable. More generally, in spite of the lowest number of classes used (in comparison with classical n-grams), correct class sequences appear to be rare events,

as they generally occur only very few times. This is shown by the histogram of figure 1. These frequencies are computed from LeM. It appears that 34% of 3-class and more than 15% of 2-class occur only once and 34% of 2-class and 62% of 3-class occur less or equal than 5 times. The high number of events seen only one time is due to the sparse data and to the errors of automatic labelling. The total number of 2-class and 3-class in the corpus are respectively 17500 and 255000. To handle the problem of sparse data, we used a technique of smoothing. There are many techniques of smoothing in the literature [4][5]. The basic idea in smoothing is rather simply using a 3-class to estimate the probability of a category c_i at position i, we use a formula that combines 3-class, 2-class, 1-class and 0-class.

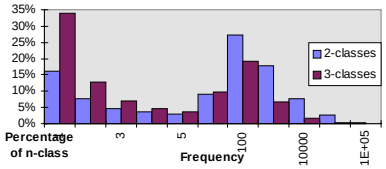

Fig. 1: Percentage of 2-class and 3-class occuring between x_i and x_{i-1} times (x-axis) in the corpus.

The probability $P(c_i / c_{i-2}c_{i-1})$ is then estimated by the formula :

$$\alpha P(c_i / c_{i-2}c_{i-1}) + \beta P(c_i / c_{i-1}) + \gamma P(c_i) + \theta \quad (7)$$

where $\alpha + \beta + \gamma + \theta = 1$. This formula guarantees a nonzero estimate for $P(c_i / c_{i-2}c_{i-1})$. The model reaches its best performance when α is significantly greater than the other parameters. To estimate these parameters, we use a technique proposed by Jelinek. We can consider the interpolated n-class language as a Markov chain which has five states: an initial state S_i and states S_3, S_2, S_1, S_0 where each S_k corresponds to a k-class model (except for Si) and the transition probabilities are respectively the parameters $\alpha = P(S_3/S_i)$, $\beta = P(S_2/S_i)$, $\gamma = P(S_1/S_i)$ and $\theta = P(S_0/S_i)$. We make some experiments to determine these parameters when the size of the training and test fluctuate. In our experiments, we computed these parameters by varying the training corpus from 1 sub-corpus until 23 and a training test varying from 23 sub-corpus to one sub-corpus.

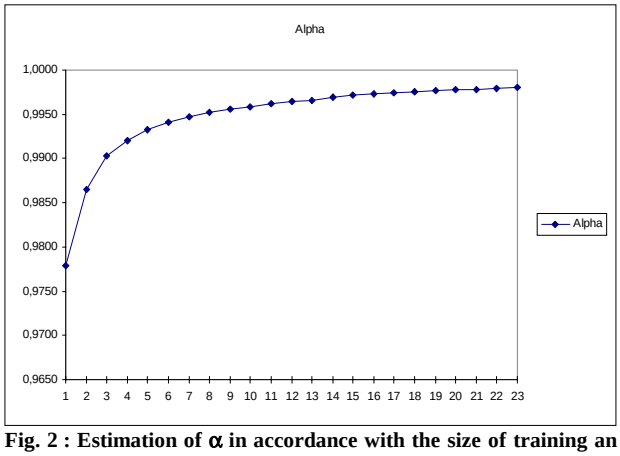

Fig. 2 : Estimation of α in accordance with the size of training an test corpus.

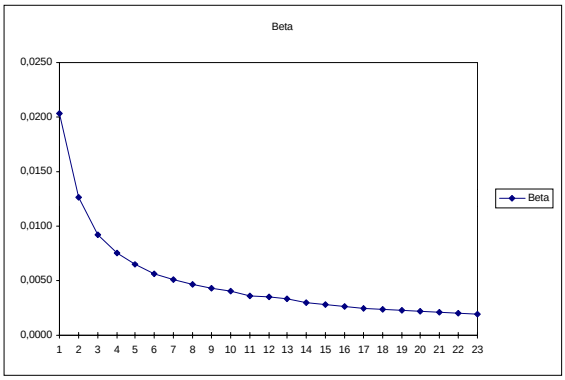

Fig. 3 : Estimation of β in accordance with the size of training an test corpus.

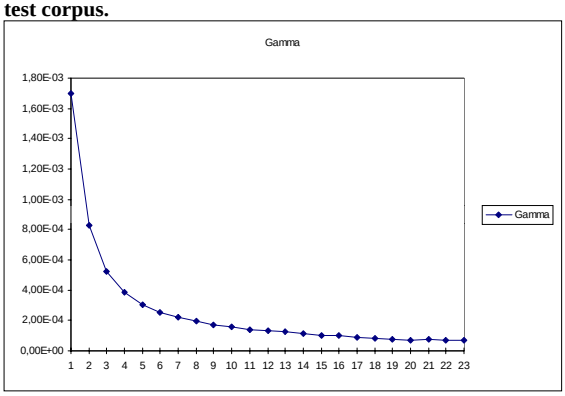

Fig. 4 : Estimation of γ in accordance with the size of training an test corpus.

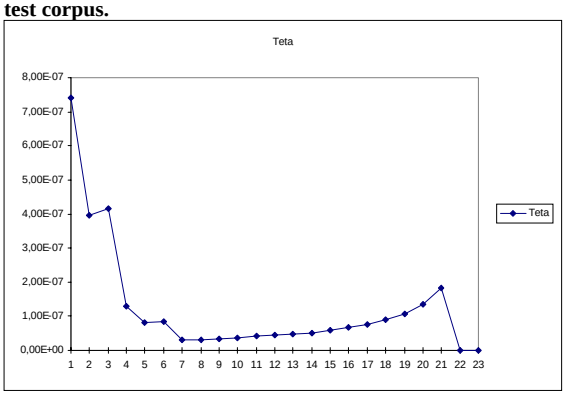

Fig. 5 : Estimation of θ in accordance with the size of training an test corpus.

The curves of figures 2, 3, 4 and five show the evolution of the n-class language model parameters in accordance with different sizes of training and test corpus. In these experiments the LeM corpus has been splited on 24 sub-corpus. The size of each corpus is about 1,75 Million words. We can point out that the value of θ is equal to zero when the training corpus is very important but if the training corpus is not sufficiently large (as in the first experiments) some 0-class events have not appeared. We recorded in this experiment 34 0-class events which never appeared in a sub-corpus of 1,75 million words.

6. Unification Grammar Language Model

One of the most drawback of the classical n-gram is the short modelling of the history well in natural spoken language the production of a word can depend on a word or a sequence of words pronounced n words before (with n>3). Because of the structural limitation of n-gram, a

speech recognition system must take into account a very high number of hypothesis. To reduce the number of hypothesis, we decided to add explicitly linguistic knowledge in order to obtain an hybrid language model. This knowledge allows to precise how to combine the linguistic information associated at each component of the sentence. For instance, some knowledge take into account the phenomena of agreement in gender and number. To capture phenomena of French language, we wrote some grammatical rules which have been modelled by a unification grammar. A unification grammar is a grammar specified as a set of constraints between feature structures. Each feature structure is made up by a set of pairs <*attribute, value*> such as each attribute takes only one value. In the example given below, we have a structure with three attributes: category *CAT*, person *PERS* and gender *GENR* whose values are *Noun, 3* and *F* respectively {CAT : Noun, PERS : 3, GENR : F}. This grammar is based on the unification of two feature structures. Two feature structures unify if there is a feature structure that is an extension of both. A feature structure F1 extends (or is more specific than) a feature structure F2 if every feature value in F1 is specified in F2. The unification operator combines the information into two feature structures with the condition that they are compatible. For our language model, we have defined a set of basic features for French such as : gender, number, person, verb form, ... Each of these features takes values in an a priori defined set. To implement this grammar, we decided to represent it as an augmented transition network (ATN). Each transition in this network can represents either a lexicon entry or a semantic-syntactic class. The unification is handled by procedures associated to each final state. If a final state is reached the model tries either to unify or to keep information for a further unification. At present, the unification grammar takes into account only the phenomena of agreement in gender and number.

7. Experiments

The language model, we propose, has been assessed thanks to two experiments carried out by the AUPELF-UREF project. The first one consists in testing the ability of the model to filter sentences generated by the smoothed trigram model (step 4 described above). In this experiment, MAUD had the task to recognise 300 sentences. These one had pronounced by different male and female speakers. For each sentence, MAUD proposed in the third step 50 hypothesis of sentences. After labelling automatically each proposal sentence, the sequence of classes and words of each sentence are treated by the unification grammar. The unification grammar has eliminated 36% of the proposed sentences. The eliminated sentences are those which are statistically not impossible but linguistically unrealisable. The second experiment deal with the Shannon game [6] which consists in recovering a word when its complete history is known. In this experiment, the idea is to propose for each truncated sentence the N most likely words which can follow the truncated sentence. As in the previous experiment, we first labelled the truncated sentences. Then, we used the probabilistic model described in section 5 to choose the best N word hypothesis. Then, the unification grammar has been used to restrain the word hypothesis that do not unify. For this experiment, we used 1000 truncated sentences extracted from *Le monde diplomatique* newspaper. For each sentence we proposed 10 000 words hypothesis. The unification grammar have acted in only 16% of sentences because of the low number of rules in our grammar. The rate of elimination is about 1,27%. This low percentage of discarding word hypothesis is due to the fact that in this experiment all the beginning of the truncated sentence is linguistically correct. In fact, in this experiment the truncated sentences are not uttered but merely written.

8. Discussion and conclusion

In this paper, we presented an hybrid language model which overcomes the limitations of the classical n-gram. This was possible by using three kind of language model : a smoothed 3-gram, a smoothed 3-class and a formal unification grammar. This last one is used to capture linguistic phenomena which can not be done with the classical statistical language models. This language model has been used in two experiments. In the first one, the results are very satisfactory but for the second one, the results are not very high. They should be improved by increasing the database of French grammatical rules.

References

[1] J. -F. Mari, D. Fohr and J. -C. Junqua « A second-Order HMM for High Performance Word and Phoneme-Based Continuous Speech Recognition », VOL 1, PP 435, Atlanta, 1996

[2] A. Krioule, J. -F. Mari and J. -P. Haton « Some Improvements in Speech Recognition Algorithms Based on HMM. Proceedings IEEE ICASSP 90, PP 545-548, Albuquerque, 1990

[3] K. Smaili, F. Charpillet and J. -P. 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

[4] H. Ney, U. Essen and R. Kneser « On Structuring Probabilistic Dependences in Stochastic Language Modelling », in Computer Speech and Language, Vol 8, PP 1-38, 1994.

[5] F. Jelinek, R. Mercer and S. Roukos « Principles of Lexical Language Modeling for Speech Recognition », in Avances in Signal Processing Furui S. New-York, Marcel Dekker, PP 651-699, 1992

[6] C.E. Shannon « Prediction and Entropy of Printed English », Bell Syst. Techn. J. PP 50-64, January 1951.