ACADEMIC PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Speech and Language 17 (2003) 27–41

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Statistical language modeling based on variable-length sequences

Imed Zitouni*,[1], Kamel Smaïli, Jean-Paul Haton

LORIA/INRIA-Lorraine, 615 rue du Jardin Botanique, BP 101, F-54600 Villers-lès-Nancy, France

Received 21 September 2000; accepted 13 May 2002

Abstract

In natural language and especially in spontaneous speech, people often group words in order to constitute phrases which become usual expressions. This is due to phonological (to make the pronunciation easier), or to semantic reasons (to remember more easily a phrase by assigning a meaning to a block of words). Classical language models do not adequately take into account such phrases. A better approach consists in modeling some word sequences as if they were individual dictionary elements. Sequences are considered as additional entries of the vocabulary, on which language models are computed.

In this paper, we present a method for automatically retrieving the most relevant phrases from a corpus of written sentences. The originality of our approach resides in the fact that the extracted phrases are obtained from a linguistically tagged corpus. Therefore, the obtained phrases are linguistically viable. To measure the contribution of classes in retrieving phrases, we have implemented the same algorithm without using classes. The class-based method outperformed by 11% the other method.

Our approach uses information theoretic criteria which insure a high statistical consistency and make the decision of selecting a potential sequence optimal in accordance with the language perplexity. We propose several variants of language model with and without word sequences. Among them, we present a model in which the trigger pairs are linguistically more significant. We show that the use of sequences decrease the word error rate and improve the normalized perplexity. For instance, the best sequence model improves the perplexity by 16%, and the the accuracy of our dictation system (MAUD) by approximately 14%. Experiments, in terms of perplexity and recognition rate, have been carried out on a vocabulary of 20,000 words extracted from a corpus of 43 million words made up of two years of the French newspaper *Le Monde*. The acoustic model (HMM) is trained with the Bref80 corpus.

* Corresponding author.
E-mail address: zitouni@research.bell-labs.com (I. Zitouni).
[1] Currently with Bell Labs Lucent Technologies.

0885-2308/03/$ - see front matter © 2002 Published by Elsevier Science Ltd.
doi:10.1016/S0885-2308(02)00026-8

1. Introduction

Statistical language models are widely used for introducing additional constraints in a speech recogniser, and hence improving its performance. The role of a language model is to estimate the prior probability of word sequences. Words are commonly used as the basic lexical units in standard language models for automatic speech recognition (ASR). Nevertheless, it is noticeable that a substantial number of short phrases have a very high frequency in natural languages. Such word sequences play an important role in the improvement of a language model and are often used as single units. Word sequences improve a language model especially by their ability to capture long contexts. Indeed, with variable-length sequences, the fixed context of language models, like n-gram or n-class, is dynamically enhanced, depending on the length of word sequences. Some sequences may have meanings which differ from those of the individual words that compose them, i.e. they constitute collocation. Such sequences (e.g. "write-off") may also have different statistical properties from the component words ("write," "off"). Sequence-based language models can also improve ASR accuracy by allowing a better phonological modeling of a word sequence (e.g. "I am going to" $\mapsto$ "I'm gonna"). Consequently, the output of the speech decoder contains more linguistic information than the word string. This is due to the fact that several word sequences often have linguistic structures which contribute to the recognition of a sentence.

To take advantage of phrases, we propose to build language models which include units made up of both single words and sequences of words. However, introducing word sequences as additional dictionary entries could make the estimation of parameters less reliable and thus deteriorate the quality of the language model. Therefore, word sequences should not be arbitrarily included in the initial vocabulary.

In this paper, we present a new approach based on syntactic classes which retrieves viable linguistic variable-length sequences from a stream of observations by reducing perplexity. These typical variable-length sequences are automatically extracted from text data by using a mutual information criterion, and this process is driven by 233 French syntactic classes. The aim of this paper is also to discuss and to evaluate the influence of these typical word sequences on the most successful language models: n-gram and n-class. We denote by n-SeqGram and n-SeqClass the extension of n-gram and n-class, respectively, based on typical word sequences. To include additional types of dependencies, we propose to extend the basic trigger model (Rosenfeld, 1994; Tillmann & Ney, 1996) in order to build trigger sequences in which the trigger and the triggered tokens are either single words or phrases. This way of building trigger pairs is linguistically more significant, as shown further below.

This paper is organized as follows. We review in Section 2 the main approaches to extracting variable-length word sequences. In Section 3, we introduce our approach, and we give the theoretical background of building variable-length word sequences. Section 4 presents the evaluated language models based on typical word sequences. Then, an evaluation of these language models in terms of perplexity and word error rate obtained with our ASR system MAUD (Fohr, Haton, Mari, Smaïli, & Zitouni, 1997; Zitouni & Smaïli, 1997) is reported in Section 5. Finally, we give in Section 6 a conclusion and some perspectives.

2. Principal variable-length sequence models

Several statistically based procedures for automatically building compound words have already been described in the literature. We recall some of them in this section.

In (Jelinek, 1990) typical sequences are created by repeated applying the concept of mutual information between two adjacent words. Two words appear as a sequence if their mutual information and their occurrence number exceed predefined thresholds. For this method, the final vocabulary may include phrases of unlimited length.

Giachin et al. determine the word sequences automatically with an optimization criterion which reduces the test set perplexity (Giachin, Baggia, & Micca, 1994; Giachin, 1995). The basic idea of this approach is to choose at each iteration the pair that best reduces the log-probability of the training class corpus, and to consider it as a candidate unit to be added to the vocabulary. This training class corpus is also used to build a new class bigram model. If the perplexity is reduced when a candidate pair is used as a lexical unit, then this pair is added as a new unit to the vocabulary. The process is repeated until perplexity stops decreasing.

Ries also uses perplexity as an optimality criterion (Ries, Buo, & Waibel, 1996). The only difference with Giachin is that, at each iteration, candidate word sequences which reduce the perplexity are extracted and are integrated into the vocabulary. (Suhm & Waibel, 1994) as well as (Kenne, O'Kane, & Pearcy, 1995) use a similar concept with the difference that they choose the class candidates according to their mutual information, instead of the log-probability.

Another approach has been proposed by Beaujard and Jardino (Beaujard & Jardino, 1999). It uses different measurements compared to those presented before, i.e. bigram occurrences, mutual information, probability of the current unit given the precedent one, and probability of the current unit given the following one. This approach starts by sorting adjacent unit couples in descending order according to one of the preceding measurements. Then, the sequences which improve the corpus likelihood are inserted and considered as entry units. This process is repeated until the probability of the corpus stops improving. The gain in performance brought by this approach was not very significant compared to a bigram model.

Deligne et al. built word sequences (n-multigrams) by optimising the likelihood of word strings (Deligne & Bimbot, 1995; Deligne & Bimbot, 1997). This likelihood is computed by summing up the likelihood values of all possible segmentations of the string into sequences of words. One can note that, due to the very large number of possible sequences extracted from a vocabulary of thousands of words, the algorithm needs intensive computation.

In the following, we will present our approach which takes advantage of most of these methods but in addition uses a linguistic set of classes to make the retrieved sequences linguistically viable.

3. Word sequence selection

3.1. Overview of the method

Considering the capability of class based approaches to cope with the sparseness of data in traditional n-gram modeling, we have explored their potential in order to retrieve important word

sequences for French (Zitouni, Mari, Smaïli, & Haton, 1999). This method is entirely automatic, and minimizes perplexity by making local optimizations. The originality of our approach lies in the use of linguistic classes to extract phrases which make them linguistically viable. We begin by tagging the corpus with a set of syntactic classes C, in which words are partitioned into equivalence classes (Smaïli, Zitouni, Charpillet, & Haton, 1997). To control the convergence of the algorithm, the maximum length of a word sequence is fixed to a value q determined experimentally. This value limits the size of class sequences and, consequently, the length of word sequences.

3.2. Algorithm

The method starts by identifying the set of word sequences obtained from the concatenation of two classes, or class sequences, that minimize the perplexity. Candidate sequences whose class mutual information are close to the maximum, and whose counts are above a given threshold, are chosen. This makes the construction of sequences which are more relevant.

Let V be the word vocabulary, and T_J the mutual information threshold:

$$T_J = p \max_{c_i \in C, c_j \in C} J(c_i, c_j). \tag{1}$$

The parameter p which is close to 1 allows the selection of only classes for which the mutual information is close to the maximum. $J(c_i, c_j)$ denotes the mutual information of the pair of adjacent classes or class sequences c_i and c_j in the training corpus. The mutual information of a pair of adjacent classes (c_i, c_j) is computed as

$$J(c_i, c_j) = \log \frac{N(c_i, c_j)N}{N(c_i)N(c_j)}, \tag{2}$$

where $N(.)$ denotes the count function, and N the size of the training corpus. A large value of $J(c_i, c_j)$ indicates that c_i and c_j occur as a sequence much more frequently than can be expected from pure chance. Let L_c and L_w be, respectively, the minimum occurrence of class sequences and word sequences, under which candidates cannot be accepted as sequences. The sequence retrieval algorithm proceeds as follows:

1. Determine all the consecutive couples c_i, c_j in the class training corpus for which the mutual information $J(c_i, c_j)$ is greater than T_J. For each of them, the length must be less than q, and its occurrence greater than L_c.
2. Label the training class corpus by using all the sequences found in (1).
3. Use the word corpus to extract from the class sequences obtained in (1) the corresponding word sequences.
4. Keep all the sequences with occurrence greater than L_w, add the set of new word sequences $\{s_i, s_j\}$ obtained at step 3 to the vocabulary and modify the word corpus accordingly.
5. Compute the perplexity on a development corpus and loop to step (1) until perplexity stops decreasing.

The perplexity is computed on a development corpus of N words according to an interpolated biclass model:

$$PP = 2^{-(1/N)\log_2 P(w_1, w_2, \ldots, w_N)}, \tag{3}$$

where $P(w_1, w_2, \ldots, w_N)$ is defined as

$$P(w_1, w_2, \ldots, w_N) = p(w_1) \prod_{i=2}^{N} p(w_i/w_{i-1}). \tag{4}$$

Since a word can belong to several classes, the conditional probability $p(w_i/w_{i-1})$ is computed as in (Cerf-Danon & El-Bèze, 1991):

$$p(w_i/w_{i-1}) = \sum_{c(w_i) \in C_{w_i}} p(w_i/c(w_i)) \sum_{c(w_{i-1}) \in C_{w_{i-1}}} p(c(w_{i-1})/w_{i-1}) p(c(w_i)/c(w_{i-1})), \tag{5}$$

where C_{w_i} denotes the set of syntactic classes which contain the word w_i, and $c(w_i)$ a possible class of w_i.

The parameter set of the biclass model is then defined by the conditional probabilities $p(w_i/c(w_i))$, $p(c(w_i)/w_i)$ and $p(c(w_i)/c(w_{i-1}))$. If we denote by $N(w_i/c(w_i))$ the count of the word w_i tagged by $c(w_i)$, the probability $p(w_i/c(w_i))$ is approximated as follows:

$$p(w_i/c(w_i)) = \frac{N(w_i/c(w_i))}{N(c(w_i))}. \tag{6}$$

The probability that the class $c(w_i)$ tags the word w_i is defined by

$$p(c(w_i)/w_i) = \frac{N(w_i/c(w_i))}{N(w_i)}. \tag{7}$$

To estimate the conditional probability $p(c(w_i)/c(w_{i-1}))$, and to avoid zero-probability, the biclass is interpolated with a uniclass and a zero-class (Smaïli et al., 1997):

$$p(c(w_i)/c(w_{i-1})) = \alpha_1 \frac{N(c(w_{i-1}), c(w_i))}{N(c(w_{i-1}))} + \alpha_2 \frac{N(c(w_i))}{n} + \alpha_3. \tag{8}$$

To speed up the perplexity computation, this formula has been suitably modified in order to take into account only the adjacent word pairs including new sequences. Consequently, the necessary amount of computation has been considerably reduced.

3.3. Convergence of the algorithm

Figure 1 shows the perplexity in relation to the number of iterations for different values of q. The algorithm reaches its optimum for a length sequence of 6 and in only 10 iterations. Since, the size of the test corpus is reduced when word sequences are replaced by single symbols, the perplexity has to be normalized; we just need to keep the original number of words (N) unchanged (Equation 3). It can easily be shown that the formula proposed in (Adda, Adda-Decker, Gauvain, & Lamel, 1997) to estimate the perplexity of a language model based on word sequences is equivalent to the baseline one (Equation 3) if we consider N as the number of words instead of the number of sequences.

It is important to note that in order to generate long word sequences (e.g. "what time is it") it is necessary to generate many shorter sequences before (e.g. "what time"). Some of these shorter sequences are no longer useful after the longer ones have been generated. They have thus to be discarded if they do not increase the perplexity when they are removed from the vocabulary.

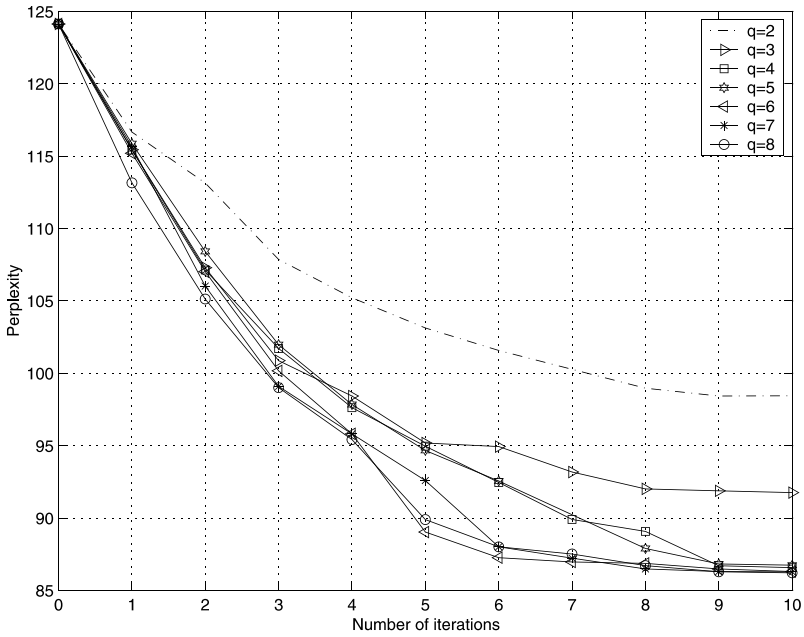

Fig. 1. Convergence of the class-based sequences algorithm. Convergence is in terms of perplexity according to the number of iterations and the maximum number of words in one sequence (q).

When words are replaced by sequences in the corpus, the priority is always given to long sequences, i.e. the longest sequences are matched first. To show the importance of using classes, we present in Fig. 2 the convergence of the algorithm (cf. Section 3.2) using only words. Results confirm our thought; the use of classes allows the algorithm to converge to a lower perplexity and hence improves its accuracy.

3.4. Characteristics of sequences

The analysis of word sequences extracted from the corpus shows that some of them have a syntactic phrase structure, whereas others are semantically valid. Table 1 presents a sample of such word sequences. Each row of the table begins with a header which contains a syntactic class sequence. The items of a row are word sequences extracted from the class sequence of the same row. For instance, the class sequence *NOM DDE NOM* (*NOM*: noun class, *DDE* a class which only contains the preposition "de" (*of*)) generates the word sequences: "chef de gouvernement" (head of government), "hommes d'affaires" (businessmen), etc. A majority of the obtained phrases correspond to syntactic forms, or carry a semantic concept. We think that this is due to the fact that the source which generates word phrases is made up of syntactic and/or semantic phrases.

3.5. Word-based sequence extraction

To measure the importance of the use of classes in retrieving sequences, we developed a sequence bigram language model where the sequences are extracted without using classes as

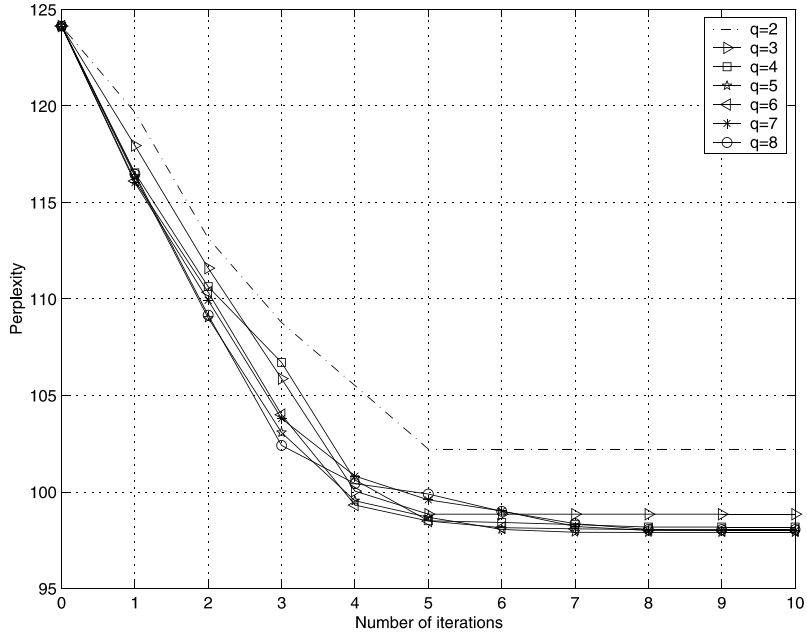

Fig. 2. Convergence of the word-sequence extraction algorithm without using classes. Convergence is in terms of perplexity according to the number of iterations and to the maximum number of words in a sequence (q).

described in Section 3.2. The obtained perplexity value is equal to 97.9, whereas the model using classes yields a perplexity value of 86.3. This result clearly shows the importance of using linguistic classes in building phrases and consequently in the design of the language model.

4. Language models based on sequences

To better understand the importance of typical word sequences, we have built a number of language models based on the classical ones: n-gram, n-class, cache, triggers. In each case the classical model has been recomputed with a new vocabulary containing not only single words, but also word sequences extracted by the algorithm presented in Section 3. Each sequence of words in the vocabulary is treated as a single entry. Thus, the classical parameter estimation methods can be used without any change. These new language models are described below. Each model uses the same training corpus which has been used to produce a sequence corpus according to the variable-length sequences built by the algorithm described in Section 3.2.

4.1. N-gram sequences: n-SeqGram

The well-known n-gram models have proved their utility in ASR, but they have also shown their limit. These models have the ability to capture local constraints, but a natural language contains structures that need more powerful models. To make this possible without increasing the fixed size of the history which is sensitive to sparse data, we proposed and developed a model

Table 1
A sample of extracted classes and their corresponding word sequences

NOM DDE NOM (Noun, Preposition, Noun): chef de gouvernement, conditions de travail, conférence de presse, fin de semaine, hommes d'affaires, juge d'instruction, milliards de francs, pouvoir d'achat,...
NEG ETR PAS (Negation form with the verb to be): n'était pas, n'est pas, ne sera pas, ne serait pas, ne soient pas, ne soit pas, ne sont pas.
CAR NOM (Cardinal numbers associated to a noun): cent ans, cinquante francs, deux jours, deux mille, un an, un exemple, un milliard, un mois,...
ARI NOM DDE (Indefinite article, Noun, Preposition): un an de, un état de, un gouvernement de, une réunion de, un langage d', un million de, une mission de, une politique de,...
PPE PCR VEC (Pronoun associated to a verb): il s'agit, il s'agissait, il se fait, il se trouve, je me sens, je me souviens, on se demande, on se trouve,...
ARD NOP (Definite article, Proper Noun): l'Afrique, l'Amérique, l'Occident, la CEE, la France, le Canada, le Monde,...
NOP NOP (Proper Noun, Proper noun): Arabie Saoudite, Edouard Balladur, George Bush, Jacques Chirac, Lionel Jospin, Los Angeles, Maison Blanche, Raymond Barre,...
Others: Alpes de Hautes Provence, au fur et à mesure, Coca Cola Côte d'Ivoire, Côte d'Or,...

called n-SeqGram. This model uses a fixed size sequence history, but, in terms of words, it is actually a variable word sequence model. In this model, the likelihood of a chain S of N sequences $S\ (s_1 \dots s_N)$ is

$$P(s_1 \dots s_N) = \prod_{i=1}^{N} P(s_i / s_{i-n+1}^{i-1}). \tag{9}$$

The parameters of the model are estimated by using the Katz method as follows:

$$P(s_i / s_{i-n+1}^{i-1}) = \begin{cases} f(s_{i-n+1}^{i}) & \text{if } N(s_{i-n+1}^{i}) > k, \\ df(s_{i-n+1}^{i}) & \text{if } 0 < N(s_{i-n+1}^{i}) \le k, \\ \alpha(s_{i-n+1}^{i-1}) P(s_i / s_{i-n+2}^{i-1}) & \text{otherwise,} \end{cases} \tag{10}$$

where $f(x)$ is the relative frequency of x, $N(x)$ the count of x, d the discounting coefficient estimated by the Good-Turing method, k the value under which the counts are unreliable, and α the back-off parameter which depends on the sequence chain $s_{i-n+1} \dots s_{i-1}$.

4.2. N-class sequences: n-SeqClass

N-class language models are usually used to cope with the problem of data sparseness. To evaluate the gain brought by sequences, we developed a new model called n-SeqClass. This model is based on classes which have been built by using explicit linguistic knowledge. The probability of a chain of sequences is given by

$$P(s_i/s_{i-n+1}^{i-1}) = P(s_i/C(s_i)) \times P(C(s_i)\dots C(s_{i-n+1})), \tag{11}$$

where $C(x)$ is the class of sequence x. Since a sequence can belong to several classes, formula (11) has to be modified as follows:

$$\begin{aligned} P(s_i/s_{i-n+1}^{i-1}) &= \sum_{c(s_i)\in C_{s_i}} p(s_i/c(s_i))p(c(s_i)/s_{i-n+1},\dots,s_{i-1}) \\ &= \sum_{c(s_i)\in C_{s_i}} p(s_i/c(s_i))\left[\sum_{c(s_{i-1})\in C_{s_{i-1}}} p(c(s_{i-1})/s_{i-1})p(c(s_i)/s_{i-n+1},\dots,s_{i-2},c(s_{i-1}))\right] \\ &= \sum_{c(s_i)\in C_{s_i}} p(s_i/c(s_i))\left(\sum_{c(s_{i-1})\in C_{s_{i-1}}} p(c(s_{i-1})/s_{i-1})\dots\right. \\ &\quad \left.\times \sum_{c(s_{i-n+1})\in C_{s_{i-n+1}}} p(c(s_{i-n+1})/s_{i-n+1})[p(c(s_i)/c(s_{i-n+1})\dots c(s_{i-1}))]\right), \end{aligned} \tag{12}$$

where $C(s_n)$ is a possible tag for s_n, C_{s_n} the set of classes to which s_n belongs, and $P(s_i/C(s_i))$ the probability that the sequence s_i should be tagged by $C(s_i)$.

4.3. Cache sequence model: SeqCache

To include distant relationships, we designed a new language model based on a cache of sequences. As for a classical cache model, this model is very useful to reinforce the probability of a sequence which has been met in a remote history. For instance, it is able to better predict *the president of the USA* if this sequence is used as a single unit in the vocabulary, whereas the classical cache predicts only the words of this sequence one by one.

For a cache of M words, the probability of a sequence is set to

$$P_{\text{SeqCache}}(s_i/s_{i-M}^{i-1}) = \frac{1}{M}\sum_{m=1}^{M}\delta(s_i,s_{i-m}), \tag{13}$$

where δ is the *Kronecker* delta symbol. Obviously, the cache is not used alone, but rather linearly interpolated with a n-SeqGram. The length of the history has been experimentally set to a cache of 50 sequences.

Table 2
A sample of trigger pairs $\{s_i \rightarrow s_j\}$

président directeur → société,	Fahd → Arabie Saoudite,
millions de barils → cours,	le gouvernement → Jacques Chirac,
cohabitation → Chirac,	Montparnasse → Paris,
Françaises → Français,	observation militaire → programme,
la Syrie → Damas,	RPR → gaullisme,
Matignon → le premier ministre,	intersyndicale → CFDT,
symphonie → orchestre,	les cheminots → grève,...

4.4. Trigger sequence model: SeqTrigger

To generalize the principle of cache sequences, and to extract more information from the long distance document history (Tillmann & Ney, 1996), we implemented a model based on triggers of sequences. In this model, a sequence of words (S_1) triggers another sequence (S_2) if both sequences are significantly correlated. That means that, if S_1 happens, S_2 will probably occur again further in the document. Consequently, the language model has to increase the probability of S_2. As for the cache sequence model, the advantage of this type of model is its capability of joining two long sequences that are syntactically or semantically close. By using mutual information, we keep the n best sequence of triggers. Experiments have shown that with more than $75K$ sequence of triggers the perplexity stops decreasing. Thus, we decided to keep the $75K$ best sequences which occurred more than 7 times. Actually, the SeqTrigger model has not been used alone, but it has been linearly combined with a n-SeqGram, and a SeqCache. Table 2 shows a set of trigger pairs extracted in accordance with the approach presented above.

5. Experimental assessment

The evaluation of the different language models based on sequences has been carried out in terms of perplexity and of ASR error rate.

5.1. Data description

The language models have been built with a French corpus (LeM), which represents 2 years (87–88) of "Le Monde" newspaper and contains 43 million words. To estimate the n-SeqClass and the n-class models, we used a set of 233 classes extracted from the 8 elementary grammatical classes of the French language including punctuation (Smaïli, Charpillet, & Haton, 1996). Training and test corpora have been tagged[2] with an algorithm based on the Viterbi approach (Smaïli et al., 1997).

The basic vocabulary is made up of the most frequent $20K$ words of the corpus. The total number of typical word sequences is approximately equal to 4000. The number of trigger pairs

[2] The necessity of tagging is due to the fact that a word can belong to different classes.

Table 3
Test perplexity of different language models with and without typical word sequences

LM based on words	PP	LM based on sequences	PP	Improvement
Bigram	121.53	SeqBigram	83.63	31%
Trigram	74.65	SeqTrigram	63.96	14%
Biclass	135.11	SeqBiclass	89.12	34%
Triclass	84.18	SeqTriclass	73.80	12%
Bigram + cache + trigger	117.53	SeqBigram + SeqCache + SeqTrigger	80.00	32%
Trigram + cache + trigger	72.69	SeqTrigram + SeqCache + SeqTrigger	60.95	16%

is about $500K$, from which we keep the $75K$ best ones. To estimate the HMM2 acoustic phone models, we use the Bref80 spoken corpus for French (Lamel, Gauvain, & Eskenazi, 1991).

5.2. Results in terms of perplexity

Perplexity (PP) is widely considered as a measure for evaluating language models. It is therefore interesting to compare the perplexity values obtained by the language models with and without typical word sequences. Tests have been carried out on a corpus of 5 million words extracted from "Le Monde" newspaper (87–88).

The language models (LM) that have been evaluated are shown in Table 3. The "back-off" method was used to estimate language models (Katz, 1987), and the range over which the discounting occurs is 7 for a trigram – 3-SeqTrigram, 7 for a bigram – 2-SeqBigram and 1 for a unigram – 1-SeqGram. No cutoff has been used in the models.

Results show that the introduction of typical word sequences greatly improves the perplexity (up to 34% for the third model). The largest improvement is obtained for language models based on a history of one unit. Indeed, the introduction of sequences in a bigram model (SeqBigram) decreases the perplexity by about 38 points. For models based on a history of two units, the improvement is between 12% and 14%. This improvement is not as large as for the preceding model. This is due to the small amount of available data. In fact, for an history of two sequences, these can be compounded by up to twelve words. We can also notice that the introduction of triggers and cache (with or without sequences) improves the perplexity by 4.7%.

We have also computed the perplexity on the corpus used for the speech recognition evaluation (cf. Section 5.3). This corpus (300 sentences) has been supplied by AUPELF-UREF[3] for the evaluation of French speech recognition systems (Dolmazon et al., 1997). Table 4 shows several language models with cutoffs indicated by (cutoff unigram–cutoff bigram–cutoff trigram) (Seymore & Rosenfeld, 1996). In each case the introduction of sequences improves the perplexity. This improvement reaches 28% for a SeqBigram with the best cutoff.

[3] An agency in charge of promotion of French language.

Table 4
Test perplexity of model cutoffs for the AUPELF corpus

Models	Cutoffs	PP
Bigram	1–7	211.54
SeqBigram	1–7	151.05
Bigram	1–5	211.76
SeqBigram	1–5	151.81
Trigram	1–7–10	153.28
SeqTrigram	1–7–10	124.32
Trigram	1–5–5	154.08
SeqTrigram	1–5–5	125.02

5.3. Results in terms of ASR

5.3.1. The MAUD system

The ASR evaluation has been done by using MAUD (Fohr et al., 1997), our continuous speech dictation system. Figure 3 shows the architecture of MAUD. The basic version of the system operates in four steps:

- Gender identification: the aim of this step is to determine the gender of the speaker by using a beam search algorithm.
- Word lattice generation: its goal is to build a word lattice from the speech signal. Context-dependent acoustic models are used according to the results of the first step. This step uses a Viterbi block algorithm which takes into account phonological alterations, and a bigram language model.
- *K*-best sentences: sentences are built from the word lattice by using a trigram language model and the acoustic scores obtained at the preceding step. A beam search algorithm is used to get the *K*-best sentences with *K* set to 80.
- Sentence filtering: a set of syntactic constraints is used in order to obtain the best sentence (Zitouni & Smaïli, 1997).

5.3.2. Acoustic model

Each phoneme is modeled by a second order hidden Markov model (Mari, Haton, & Kriouile, 1997) with 3 states (HMM2). Each single word in the vocabulary is represented by the concat-

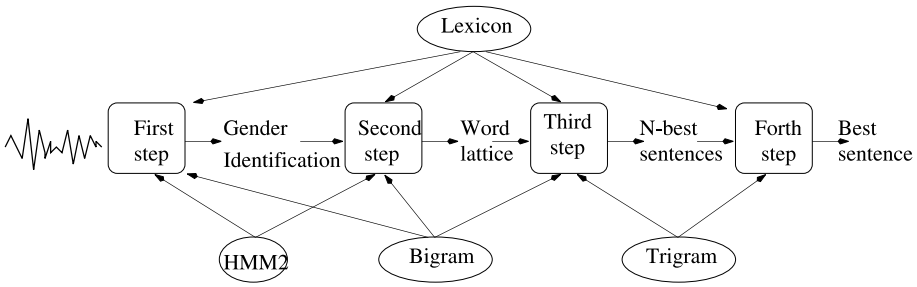

Fig. 3. The architecture of the MAUD dictation machine.

Table 5
Recognition results of different versions of MAUD with and without typical word sequences

Rate	*VM*1	*VS*1	*VM*2	*VS*2	*VM*3	*VS*3
Correct (%)	60.2	70.1	54.7	66.5	61.7	71.9
Substitution (%)	31.0	23.5	35.5	26.4	29.7	22.1
Deletion (%)	8.8	6.4	9.8	7.1	8.6	5.9
Insertion (%)	5.9	6.1	6.0	5.8	6.5	6.8
Accuracy	54.3	64.0	48.7	60.7	55.2	65.1
WER	45.7	36.0	51.3	39.3	44.8	34.8

enation of the HMM2 phones which compose it. If the vocabulary unit is a typical word sequence, we introduce an optional HMM2 silence phone between the single words that compose it. Thus, for two adjacent words A and B in a sequence, we evaluate on a training corpus the transition probabilities between the ending HMM2 phone of A, the HMM2 silence phone, and the beginning HMM2 phone of B.

5.3.3. Recognition results

Several versions of MAUD have been considered in order to evaluate the gain brought by the introduction of typical word sequences, i.e.:

- *VM*1 which is the base version presented above without word sequences;
- *VS*1 which uses a 2-SeqGram in the second step and a 3-SeqGram in the third step (with typical word sequences);
- *VM*2, similar to *VM*1, with the difference that biclass and triclass are used instead of bigram and trigram, respectively;
- *VS*2, in which we introduce typical word sequences and we replace the biclass and the triclass models by the 2-SeqClass and the 3-SeqClass, respectively;
- *VM*3, in which we add single word triggers and cache models in the third step of the *VM*1 version;
- *VS*3, in which we add word sequence triggers and cache models in the third step of the *VS*1 version.

A summary of results is presented in Table 5. In this table, the rate of words predicted correctly (correct rate) are given together with the substitution, deletion and insertion rates. The accuracy is defined as the correct rate from which the insertion rate is subtracted. The word error rate (WER) is equal to the summation of substitution, deletion and insertion rates. These rates are computed according to the NIST[4] toolkit (Pallett et al., 1994; Pallett, 1997).

All experiments have been carried out on the test corpus (300 sentences) provided by AUPELF-UREF for the French ASR evaluation programme. Results show that the introduction of typical word sequences in recognition improves the word error rate (WER) and the accuracy of MAUD. Indeed, the introduction of word sequences in the basic version *VM*1 ($Accuracy = 54.3\%$) improves the recognition by 14%, by 18% for the *VM*2 version, and by 15% for the VS3 version. The use of sequences also improves the rate of correctly words predicted by 14% (7 over 10 are

[4] National Institute of Standards and Technology.

recognized), the substitution rate by 24%, and the deletion rate by 18% approximately, whereas the insertion rate is decreased by approximately 3%.

6. Conclusion and perspectives

We have presented in this paper an approach to design language models for ASR based on typical variable-length word sequences. Typical word sequences are automatically determined by an algorithm based on a perplexity minimization combined with a mutual information criterion. A gain of 16% in terms of perplexity and 14% in terms of ASR accuracy have been obtained for French. This work demonstrates the value of using variable-length sequences. One of the reasons of these good results is the use of class sequences. Such sequences are highly correlated, and, in addition, they are linguistically viable. We have compared this method with a method based on retrieving sequences only from word data, and the results were 11% worse. This shows the importance of using classes. It is worth noticing that most sequences obtained with our method are meaningful. Several are syntactic groups, and some of them have a semantic nature.

Another way to build linguistically more significant trigger pairs, has also been proposed. In contrast with the commonly used trigger approach based only on single words, our method is based on variable-length word sequences for the trigger and the triggered units.

The notion of sequences allows to take into account longer contexts of variable length. Experimental results prove that sequences allow better modeling of natural language.

References

Adda, G., Adda-Decker, M., Gauvain, J.L., Lamel, L., 1997. Text normalization and speech recognition in French. In: Proceeding of the European conference on speech communication and technology, Rhodes, Greece, pp. 2711–2714.

Beaujard, C., Jardino, M., 1999. Language modeling based on automatic word concatenations. In: Proceeding of the European conference on speech communication and technology, Budapest, Hungary, pp. 1563–1566.

Cerf-Danon, H., El-Bèze, M., 1991. Three different probabilistic language models: comparison and combination. In: Proceeding of the international conference on acoustics, speech and signal processing, Toronto, Canada, Vol. 1, pp. 297–300.

Deligne, S., Bimbot, F., 1995. Language modeling by variable-length sequences: theoretical formulation and evaluation of multigrams. In: Proceedings of the international conference on acoustics, speech and signal processing, Detroit, MI, USA, pp. 169–172.

Deligne, S., Bimbot, F., 1997. Inference of variable-length linguistic and acoustic units by multigrams. Speech Communication 23, 223–241.

Dolmazon, J.M., Bimbot, F., Adda, G., El-Bèze, M., Caërou, J.C., Zeiliger, J., Adda-Decker, M., 1997. Organisation de la première campagne AUPELF pour l'évaluation des systèmes de dictée vocale. In: Actes des premières JST Francil 1997 Avignon, France, pp. 13–18.

Fohr, D., Haton, J.P., Mari, J.F., Smaïli, K., Zitouni, I., 1997. MAUD: un prototype de machine à dicter vocale. In: Actes des premières JST Francil 1997, Avignon, France, pp. 25–30.

Giachin, E., 1995. Phrase bigrams for continuous speech recognition. In: Proceedings of the international conference on acoustics, speech and signal processing, Detroit, MI, USA, pp. 225–228.

Giachin, E., Baggia, P., Micca, G., 1994. Language models for spontaneous speech recognition: a bootstrap method for learning phrase bigrams. In: Proceeding of international conference on spoken language processing, Yokohama, Japan, pp. 843–846.

Jelinek, F., 1990. Self-organized language modeling for speech recognition. In: Waibel, A., Lee, K-F. (Eds.), Readings in Speech Recognition. Morgan Kaufmann, San Mateo, Calif, pp. 450–506.

Katz, S.M., 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics Speech and Signal Processing 35 (3), 400–401.

Kenne, P.E., O'Kane, M., Pearcy, H.G., 1995. Language modeling of spontaneous speech in a court context. In: Proceeding of the European conference on speech communication and technology, Madrid, Spain, pp. 1801–1804.

Lamel, L., Gauvain, J.L., Eskenazi, M., 1991. BREF, a large vocabulary spoken corpus for french. In: Proceeding of European conference on speech communication and technology, Gênes, Italy, pp. 505–508.

Mari, J.F., Haton, J.P., Kriouile, A., 1997. Automatic word recognition based on second-order hidden markov models. IEEE Transactions on Speech an Audio Processing 2 (1), 22–25.

Pallett, D., NIST spoken natural language processing group: benchmark tests description and public software. Technical report, http://www.itl.nist.gov/div894/894.01.

Pallett, D., Fiscus, J., Fisher, W, Garofolo, J., Lund, B., Prysbocki, M., 1994. 1993 benchmark tests fot the ARPA spoken language program. In: Proceedings of the ARPA, pp. 49–79.

Ries, K., Buo, F.D., Waibel, A., 1996. Class phrase models for language modeling. In: Proceeding of international conference on spoken language processing, Philadelphia, PA, USA, pp. 398–401.

Rosenfeld, R., 1994. Adaptive statistical language modeling: a maximum entropy approach. PhD Thesis, School of Computer Science Carnegie Mellon University, Pittsburgh.

Seymore, K., Rosenfeld, R., 1996. Scalable trigram backoff language models. Technical report, Carnegie Mellon University Tech Report CMU-CS-96-139.

Smaïli, K., Charpillet, F., Haton, J.P., 1996. 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.

Smaïli, K., Zitouni, I., Charpillet, F., Haton, J.P., 1997. A hybrid language model for a continuous dictation prototype. In: Proceeding of the European conference on speech communication and technology, Rhodes, Greece, pp. 2755–2758.

Suhm, B., Waibel, A., 1994. Towards better language models for spontaneous speech. In: Proceeding of international conference on spoken language processing, Yokohama, Japan, pp. 831–834.

Tillmann, C., Ney, H., 1996. Selection criteria for word trigger pairs in language modeling. In: Miclet, L., de la Higuera, C. (Eds.), Grammatical inference: learning syntax from sentences, Third international colloquium, ICGI-96, Montpellier, France. Lecture notes in artificial intelligence, Vol. 1147. Springer, pp. 98–106.

Zitouni, I., Mari, J.F., Smaïli, K., Haton, J.P., 1999. Variable-length sequence language model for large vocabulary continuous dictation machine: the n-seqgram approach. In: Proceeding of the European conference on speech communication and technology, Budapest, Hungary, pp. 1811–1814.

Zitouni, I., Smaïli, K., 1997. Apport d'une grammaire d'unification dans un système de dicté automatique. In: Deuxièmes journées jeunes chercheurs en parole, La Rochelle, France.