HAL
open science

Beyond the Conventional Statistical Language Models: The Variable-Length Sequences Approach

Imed Zitouni, Kamel Smaïli, Jean-Paul Haton

To cite this version:

Imed Zitouni, Kamel Smaïli, Jean-Paul Haton. Beyond the Conventional Statistical Language Models: The Variable-Length Sequences Approach. International Conference on Speech Language Processing, 2000, Pékin, China. pp.4. inria-00099107

HAL Id: inria-00099107

https://inria.hal.science/inria-00099107v1

Submitted on 21 Nov 2017

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.

BEYOND THE CONVENTIONAL STATISTICAL LANGUAGE MODELS: THE VARIABLE-LENGTH SEQUENCES APPROACH

I. Zitouni, K.Smaïli, J-P. Haton

LORIA/INRIA-Lorraine
B.P.239 54506 Nancy, France
E-mail: {zitouni, smaili, jph}@loria.fr

ABSTRACT

In natural language, several sequences of words are very frequent. A classical language model, like n-gram, does not adequately take into account such sequences, because it underestimates their probabilities. A better approach consists in modelling word sequences as if they were individual dictionary elements. Sequences are considered as additional entries of the word lexicon, on which language models are computed. In this paper, we present an original method for automatically determining the most important phrases in corpora. This method is based on information theoretic criteria, which insure a high statistical consistency, and on French grammatical classes which include additional type of linguistic dependencies. In addition, the perplexity is used in order to make the decision of selecting a potential sequence more accurate. We propose also several variants of language models with and without word sequences. Among them, we present a model in which the trigger pairs are more significant linguistically. The originality of this model, compared with the commonly used trigger approaches, is the use of word sequences to estimate the trigger pair without limiting itself to single words. Experimental tests, in terms of perplexity and recognition rate, are carried out on a vocabulary of *20000* words and a corpus of *43* million words. The use of word sequences proposed by our algorithm reduces perplexity by more than 16% compared to those, which are limited to single words. The introduction of these word sequences in our dictation machine improves the accuracy by approximately 15%.

1. INTRODUCTION

The role of a statistical language model is to estimate the prior probability of the word sequences occurring in the task. In speech recognition, words are commonly used as the basic lexical units. Nevertheless, a consistent number of short phrases have a very high frequency. To take advantage of this fact, we propose to build a language model which bundles sequences of words, which are extracted from, frequent phrases. The tokens can be both single words and sequences of words. However, introducing word sequences as additional dictionary entries makes the problem of sparseness data more crucial and thus deteriorates the language model. Therefore, word sequences must not be arbitrarily included in the initial vocabulary.

One way that word sequences improve the language model is by capturing longer 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 that differ from those of the individual words. Such sequences (e.g. "write-off") may have different statistical properties from the component words ("write", "off"). Sequence-based language models may also improve automatic speech recognition (ASR) accuracy by allowing a better acoustic modelling of inter-word boundaries (e.g. "in-the" or "you-all") and the utilisation of inter-word pronunciation variants. The output of the speech decoder contains consequently 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.

We present in this paper a new approach that aims at retrieving sequential variable-length regularities within streams of observations by reducing perplexity. These typical variable-length sequences are automatically extracted from text data, by using mutual information criterion. One of the originalities of our approach is the use of linguistic dependencies obtained by French syntactic classes. This approach aims at building variable-length sequences of words drawn from a large vocabulary (20000 words).

The purpose of this paper is also to discuss and to evaluate the performances brought by these typical word sequences on the most successful approach: n-gram and n-class. We denote by n-SeqGram and n-SeqClass the extension brought to n-gram and n-class respectively, by the set of typical word sequences. To include additional types of dependencies, we propose the idea of a trigger pair as the basic information-bearing element. If a word sequence A is significantly correlated with another word sequence B, then (A→B) is considered as a “trigger pair”, with A being the trigger and B the triggered sequence. When A occurs in the document, it triggers B, causing its probability estimate to change. The originality of this method, compared to the commonly used trigger approaches, is the use of word sequences to estimate the “trigger pair” without limiting itself to single words.

2. PRINCIPAL VARIABLE-LENGTH SEQUENCE MODELS

Several statistical-based procedures building automatically compound words have already been described in the literature.

Mercer creates typical sequences based on the concept of mutual information between two adjacent words [1]. Two words are considered as a sequence if their mutual information and occurrence number are both greater than predefined thresholds.

Giachin suggests to determine the word sequences automatically with an optimisation criterion, which reduces perplexity [2]. The basic idea of this approach is to choose at each iteration, the pair of words that best reduces the log-probability of the training class corpus. Then, this one is kept as a candidate. If the perplexity is reduced when the candidate pair is used as a unit, then this one is added as a unit in the vocabulary. The process is repeated until perplexity stop decreasing. Ries also uses perplexity as an optimisation criterion [3]. The only difference with Giachin is that Ries extracts, at each iteration, a set of candidate word sequences and integrates into its vocabulary only those sequences that reduce perplexity.

Suhm [4] as well as Kenne [5] use the same concept suggested by Giachin, with the difference that he chooses the class candidates according to their mutual information, instead of probability.

Beaujard and Jardino in [6] use different measurements compared to those presented before: bigram counts, mutual information, probability of the current unit given the precedent one and the 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 added to the dictionary. This process is repeated until the corpus liklihood stops improving.

The weakness of the above mentioned methods is their complexity. Therefore, they have been used only on few hundred words vocabularies.

Deligne builds word sequences (n-multigrams) by optimising the likelihood of word strings [7]. The adjacent words likelihood is computed by summing up the likelihood values of all possible sentence segmentations. Note that the huge number of possible sequences built from a vocabulary of thousands of words requires intensive computation.

3. WORD SEQUENCES SELECTION

Considering the success of class based approaches to cope with the sparseness of data in traditional n-gram modelling, we have explored their potential in our method [8]. This one is entirely automatic and minimises the perplexity by making local optimisations. We begin by tagging the corpus with a set of syntactic classes *C*, where words are partitioned into manually determined equivalence classes [9]. After fixing the maximum length of a word sequence *q*, the model starts by identifying the set of word sequences obtained by the concatenation of two classes or class sequences that produce a perplexity reduction. We choose all the candidate sequences whose class mutual information is close to the maximum in the corpus and whose count is above a given threshold. Let *V* be the word vocabulary and T_J be the threshold of the mutual information:

$$T_j = p \max_{c_i \in C, c_j \in C} J(c_i, c_j) \quad (1)$$

where *p* denotes the coefficient used to compute T_J, and $J(c_i, c_j)$ denotes the mutual information of the class couple or the class

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

sequences on the training corpus:

where *N(.)* denotes the count function and *T* denotes 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 T_{min} and T_{occ} be the minimum count of a candidate class and word sequence respectively. Then, the procedure proceeds as follows:

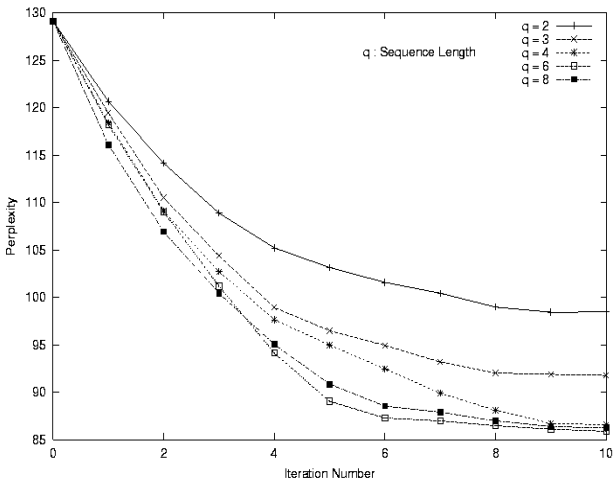
1. Determine, on the training corpus, the couples c_i,c_j for which the mutual information $J(c_i,c_j)$ is greater than T_J. The total number of classes in each couple of classes or class sequences should be less than *q*;
2. Add the set of new class sequences obtained in 1 to the class vocabulary, building *C'*, and label the class corpus accordingly;
3. Use the word corpus and the corresponding class corpus labelled by C', to extract the corresponding word sequences;
4. Add the set of new sequences $\{s_i,s_j\}$ obtained in 3 with occurrence greater than T_{occ}, to the vocabulary and modify the corpus accordingly;
5. Repeat until perplexity doesn't decrease.

Though the perplexity is computed on a "shrunk" corpus, when some phrases have been replaced by single symbols (word sequences), we have to keep the original number of words, unchanged because it is the actual number of words in the corpus [10].

It is important to note that, in order to generate long word sequences (e.g., "what time is it"); many shorter sequences have to be generated before (e.g., "what time"). Some of these shorter sequences are no longer useful after the longer ones have been generated, so they have to be discarded and their original component words should be used instead. However, in our approach we discard all shorter sequences that decrease the test perplexity when their original component words are used instead.

Figure 1 presents the convergence, in terms of perplexity, of the procedure cited above, according to iterations, and number of words in a sequence. The results show that the procedure reaches its optimum for a value of *q* equal to *6* in only *10* iterations.

Figure 1: convergence in terms of perplexity of the algorithm,



according to the number of iterations and the length of sequences.

4. EXTENSION OF BASIC LANGUAGE MODELS

4.1 The n-SeqGram and the n-SeqClass Models

In automatic speech recognition, the most widely used and successful language model are the so-called n-gram and n-class models, where the dependency of the word under consideration is limited to the immediate predecessor words. To evaluate the performance brought by the typical word sequences extracted by the algorithm presented above, we developed a n-gram and a n-class models on an extended vocabulary that include both single words (which are still necessary to model the less frequent phrases) and word sequences. In other words, typical word sequences are treated as they were single lexicon entries. We denote by n-SeqGram and n-SeqClass the extension brought to n-gram and n-class respectively.

4.2 Sequence Trigger Based Modelling

It is clear that several sort of long-distance dependencies exist as well. To include long-distance dependencies in language models, we propose to use trigger pairs. Unlike the commonly approaches presented in the literature, where *A* and *B* are restricted to single words [11], the selection criterion used in this paper is based on trigger pairs where both the triggered and the triggering events are single words or word sequences.

A natural measure of the information provided by *A* on *B* is their average mutual information. We used this measure to extract the *K* best trigger pairs that reduce the perplexity of the language model. The value of *K* was estimated experimentally on a test corpus.

As usual, triggers are used as an additional component to a basic language model, like n-gram. Hence, trigger pairs are advantageous by the further information they provide to a basic language model.

The language model we use is a linear interpolation between a n-SeqGram, a cache and a trigger models. To build the trigger and the cache models, we use the same principle as in [11].

5. EVALUATION

To evaluate our model in real conditions, we obviously carried out experiments in terms of perplexity and implemented it in our dictation machine MAUD. In the following, we give a brief overview of its recognizer and describe the different.

5.1 Data Description

To build language models, we use a French corpus (LeM) which represents 2 years (*87-88*) of "Le Monde" newspaper (*43* million words). To estimate the n-SeqClass and the n-class models, we use a set of *233* French syntactic classes [9]. The test and training corpora used in this approach are by a set of *233* classes [12]. It is important to note that a word can belong to different classes (ex: the word "orange" can be a noun or an adjective). The vocabulary is compounded of the most frequent *20000* words of LeM corpus. The number of typical word sequences is approximately equal to *4000*. The number of pair triggers is estimated to *500000*. To estimate the HMM2 phones, we use Bref80 spoken corpus [13].

5.2 Acoustic Model

Each phoneme is modelled by 3 states second order Markov model (HMM2)[14]. Thus, each single word in the vocabulary is represented by the concatenation of the HMM2 phones which compose it. If the vocabulary unit is a typical word sequence, we introduce an optional HMM2 silence phone between single words, which compose it. Thus, for each 2 adjacent words *A* and *B* in a sequence, we evaluate on a training corpus the transition probabilities between the output HMM2 phones of *A*, the HMM2 silence phone and the input HMM2 phones of *B*.

5.3 Perplexity Results

Perplexity is usually considered as a performance measure of language models. It is therefore interesting to look at the test perplexity values obtained by the language models with and without typical word. The test corpus (≈*5* million words) on which the perplexity was computed does not appear in the training corpus.

The language models evaluated in terms of perplexity are partitioned on 2 sets: language models based on single words **S1** and their corresponding models using typical word sequences **S2**. The S1 set includes bigram (**P1**), trigram (**P2**), biclass (**P3**), triclass (**P4**), linear interpolation between bigram, cache and single word triggers (**P5**), and linear interpolation between trigram, cache and single word triggers (**P6**). The S2 set includes 2-SeqGram (**PS1**), 3-SeqGram (**PS2**), 2-SeqClass (**PS3**), 3-SeqClass (**PS4**), linear interpolation between 2-SeqGram, cache and word sequences triggers (**PS5**), and linear interpolation between 3-SeqGram, cache and word sequences triggers (**PS6**). We use the "back-off" method to estimate language models [15]. A summary of test perplexity results is presented in Table 1.

S1	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**
PP	*121.53*	*74.65*	*135.11*	*84.18*	*117.53*	*72.69*
S2	**PS1**	**PS2**	**PS3**	**PS4**	**PS5**	**PS6**
PP	*83.63*	*63.96*	*89.12*	*73.80*	*80.00*	*60.95*

Table 1: Test perplexity of different language models with and without typical word sequences.

A comparison between these language models, in terms of perplexity, shows that each time, the introduction of typical word sequences has been done, it outperforms the basic model. For instance, the 3-Seqgram improves the trigram by *16,7%.*

5.4 MAUD System and Recognition Results

An evaluation was also done with MAUD [16], a continuous dictation system using a stochastic language model. The basic version of MAUD works in *4* steps: gender identification; word lattice generation by means of a Viterbi block algorithm and a bigram model; N-best sentences extraction by using a beam search according to combined score of the acoustic and the trigram language models; and finally sentence filtering by means of syntactic constraints in order to obtain the best sentence. This version has participated to the AUPELF-UREF campaign of dictation machine evaluation for French, and came in second place.

To evaluate the performance brought by the introduction of typical word sequences, we considered several versions of MAUD system: **M1** which is the base version presented above without word sequences; **MS1** which uses a 2-SeqGram in the second step and a 3-SeqGram in the third step (with typical word sequences); **M2** which is similar to M1 with the difference that we use biclass and triclass instead of bigram and trigram respectively; **MS2** 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; **M3** in which we add single word triggers and cache models to the third step of the M1 version, and **MS3** in which we add word sequences triggers and cache models to the third step of the MS1 version of MAUD.

A summary of recognition results (accuracy) is presented in Table 2. In these experiments, the recognition is done on the *300* test sentences delivered by AUPELF-UREF for the evaluation campaign.

	M1	**MS1**	**M2**	**MS2**	**M3**	**MS3**
Acc.	*54.3%*	*64.0%*	*48.7%*	*60.7%*	*55.2%*	*65.1%*

Table 2: Accuracy (Acc.) of different versions of MAUD system with and without typical word sequences.

Results show that the introduction of typical word sequences in recognition improves the accuracy of MAUD. Indeed, the introduction of word sequences to the basic version M1 (Acc=*54.3%*), improves the accuracy by *14%.* These sequences introduced to the M2 version (Acc=*48.7%*) improves the recognition by *18%.* Whereas, the introduction of word sequence triggers and cache to M3 (*55.2%*) improves the accuracy by 15*%*: the MS3 version (*65.1%*).

6. CONCLUSION AND PERSPECTIVES

We proposed in this paper an approach to overcome the limit of classical language models. This approach is based on typical variable-length word sequences as well as on single words. Typical word sequences to be modelled are automatically determined by a procedure that follows a perplexity minimisation combined with mutual information criterion. Test perplexity achieved more than *16%* reduction and *15%* accuracy improvement over language models based on single words.

Very interesting statement can be done about the nature of the discovered sequences. Some are merely group of words that frequently occur in a corpus. Most of them, however, are sensible word sequences representing linguistic constituents. For instance, several are syntactic groups, few of them have a semantic nature, etc.

Another manner to build "trigger pairs", linguistically more significant, has been proposed. Compared with the commonly used trigger approaches based only on single words, the trigger *A* and the triggered *B* units in the model we propose can be a variable-length word sequence.

To improve the performance of this approach, we propose to combine it with the multigram approach [7]. It also seems interesting to investigate the application of this approach to other problems: e.g., looking for semantic equivalence classes between word sequences in view of tagging concept and speech to speech automatic translation.

7. REFERENCES

[1] F. Jelinek "Self-Organized Language Modelling for Speech Recognition". In Readings in Speech Recognition, pp. 450-506. Ed. A. Waibel and K. F. Lee. Morgan Kaufmann, 1989.

[2] E. Giachin "Phrase Bigrams for Continuous Speech Recognition". ICASSP95, Detroit, pp. 225-228, 1995.

[3] K. Ries, FD. Buo and A. Waibel "Class Phrase models for Language Modelling". ICSLP96, Philadelphia, PA, USA, pp. 398-401, 1996.

[4] B. Suhm and A. Waibel. "Towards Better Language Models for Spontaneous Speech". ICSLP94, Yokohama, pp. 831-834, 1994.

[5] P.E. Kenne, M. O'Kane and HG. Pearcy. Language Modeling of Spontaneous Speech in a Court Context. EUROSPEECH95, Spain, pp. 1801-1804, 1995.

[6] C. Beaujard and M. Jardino. Language Modeling Based on Automatic Word Concatenations. EUROSPEECH99, Budapest, pp.1563-1566, 1999.

[7] S. Deligne and F. Bimbot "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams". ICASSP95, Detroit, pp. 169-172, 1995.

[8] I. Zitouni, JF. Mari, K. Smaili and JP. Haton, Variable-Length Sequence Language Model for Large Vocabulary Continuous Dictation Machine: The n-SeqGram Approach, EUROSPEECH99, Budapest, pp. 1811-1814, 1999.

[9] K. Smaïli, I. Zitouni, F. Charpillet and J. P. Haton "An Hybrid Language Model for a Continuous Dictation Prototype". Eurospeech97, Rhodes, pp. 2723-2726, 1997.

[10] G. Adda and al., "Text Normalisation and Speech Recognition in French". Eurospeech97, Rhodes, 1997.

[11] C. Tillmann, H. Ney , "Selection Criteria for Word Trigger Pairs in Language Modeling". In Grammatical Inference: Learning Syntax from Sentences, Springer, 1996.

[12] K. Smaïli, 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, 1996.

[13] L. Lamel, J. L. Gauvain, M. Eskenazi "BREF: a Large Vocabulary Spoken Corpus for French". Eurospeech91, Gênes, 1991.

[14] J. F. Mari, J. P. Haton, A. Kriouile "Automatic Word Recognition Based on Second-Order Hidden Markov Models". IEEE Trans. ASSP, 2(1), pp. 22-25, 1997.

[15] M. Katz "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer". IEEE Trans. ASSP, 35 (3), pp. 400-401, 1987.

[16] D. Fohr, J.P. Haton, J. F. Mari, K. Smaïli, I. Zitouni "MAUD: Un prototype de machine à dicter vocale". Actes 1ères JST FRANCIL, Avignon, pp. 25-30, 1997.