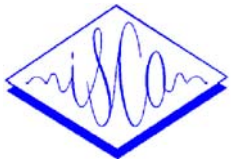

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

AUTOMATIC AND MANUAL CLUSTERING FOR LARGE VOCABULARY SPEECH RECOGNITION: A COMPARATIVE STUDY

K. Smaïli, A. Brun, I. Zitouni and J.P. Haton
{smaili, brun, zitouni, jph}@loria.fr
Loria BP 239 54506 Vandoeuvre-Lès-Nancy, France
Tel: (33) 03-83-59-20-74, Fax: (33) 03-83-41-30-79

ABSTRACT

This article describes a comparative study of language models in which the evaluation protocol has been set by AUPELF-UREF[1]. We especially pay attention on the comparison between two methods of clustering words which are necessary in the design of the corresponding language models. The first classification is done by following a linguistic and theoretical method and the second one is based on an optimization method. Both methods are evaluated through the Shannon game. The vocabulary used is 20 000 words, the training corpus is made of two years of *Le Monde* newspaper (42M of words) and the test corpus (400 000 words) is extracted from 6 years *of Le Monde Diplomatique*. First evaluations show an improvement of 13% of recognized words in the first five ranks and a decrease of 25% in perplexity.

1. INTRODUCTION

This paper deals with the problem of clustering words in order to build an efficient language model for large speech recognition systems. For more than ten years, we defend the idea that stochastic language models cannot be based only on n-grams, but have to use n-classes in order to catch more general syntactic structures. Consequently, clusters of words have to be defined. For that, one can either construct them by hand or by an automatic method. The first one is generally based on linguistic rules whereas the second depends on an iterative algorithm, whose aim is the minimization a defined measure (perplexity).

We describe here both manual and automatic word clustering processes, review the Shannon's game, present the language model which is used in research of missing words in the truncated sentences and evaluate both language models based on these classifications through the Shannon's game.

[1] Association des Universités Partiellement ou Entièrement de Langue Française - Université des Réseaux d'Expression Française.

2. THE CLUSTERING PROCESS

The clustering process is the operation, which consists in splitting the vocabulary into several sub-vocabularies.
One word can belong to more than one class. To discover the different classes, one can either compute them or construct them by hand. In both cases, one way to evaluate the likelihood of a string of words w_1 ... w_n is to compute the quantity:

$$Q = P(w_1 / C_1)\prod_{i=2}^{n} P(C_{i-1}C_i)P(w_i / C_i)$$

where C_{i-1} and C_i are respectively the classes which w_{i-1} and w_i belong to. The formula above is correct if each word has only one class label. Otherwise, if a word can be found in several classes, the Viterbi algorithm is used to determine the maximum value Q computed for each class which a word w_i belongs to.

3. THE HAND CLUSTERING

The theoretical rule used to cluster the vocabulary is the following one *: A word is put in a class if any other word of the same class can be substituted to it in a context without any change in the syntactic structure of the sentence*. To make that possible, we defined some syntactic features (contexts) and tried all the words of the dictionary on these contexts. The words that can appear in the same contexts are arranged in the same class. By applying the previous rule and refining the eight basic French grammatical classes (Nouns, Verbs, ...), we constructed 200 classes. The set of clusters is compounded of two categories: the opened and the closed classes. The opened classes are made up of the words, which could be formed, from the root of words such as verbs, adjectives, etc. The closed classes are made up of a finite number of words such as articles, pronouns, etc.
In this classification a word can belong to one or several classes, for instance the word *Le* (*The*) belongs to two classes : article and pronoun.

4. THE AUTOMATIC CLUSTERING

It is very difficult to consider a new classification for each new application domain. Therefore, the best way to adapt the language model to a new application is to learn automatically the classes, which compound it. The

lexical clustering can be viewed as a combinatorial problem: from a set of N words, how to arrange them in classes in order to obtain a reliable language model. Such a language model should give a low perplexity. There are several techniques to solve this problem [1][2], the one we use in this paper is similar to that used in [1]. This algorithm is based on the simulated annealing (SA) principle. The concept of SA is inspired from the physical annealing process of solids and is easily adaptable to solve large combinatorial optimisation problems. In condensed matter physics, people are interested in obtaining low energy states of a solid, in other words, how to arrange the billions of particles in order to achieve a highly structured lattice with a low energy of the system. The aim in automatic classification is similar. We have to arrange words in classes in order to reduce the perplexity of the language model. Let us briefly review the SA algorithm.

1.Start with a high temperature T
2.With a temperature T and until the equilibrium is reached do
3.From the current temperature T of the system and from the current state i of the system which has an Energy Ei, make a perturbation which transforms state i into state j. The energy of state j is Ej
4.If Ej-Ei ≤ 0 then state j is accepted as the current state; otherwise state j is accepted with a probability:

$$P = Min\left(1, \exp\left(\frac{E_i - E_j}{T}\right)\right)$$

5.Change the temperature and go to step 2 until the low temperature is reached

This algorithm permits to the simulation process to be released from a track of a local minimum by doing some transitions with higher energy.

5. SIMULATED ANNEALING PARAMETERS FOR WORD CLUSTERING

Seven parameters are necessary to make the above algorithm useful in word clustering, i.e., initial temperature, initial clustering, system perturbation, equilibrium state, schedule annealing, stop criterion and energy.

1. Initial Temperature

The temperature in automatic word classification will act as a control parameter. We made several experiments in order to determine the best value of the initial temperature [3]. T = 0.3 seems to be a convenient initial temperature.

2. Initial Configuration (Initial Clustering)

In the two classifications we experimented, the initial set of classes has been set either by the hand or by the random clusters.

3. System Perturbation

Moving one word chosen randomly from its class to another also selected randomly simulates the perturbation. In [3] we presented another way to move words by using contextual knowledge.

4. Equilibrium State

In our experiments, the equilibrium state has been simulated in two ways. The first one consists in making *n* perturbations before decreasing the temperature. In other words, at each temperature, *n* words are selected to move from one class to another. The second one is more interesting, because it depends on the number of accepted new configurations (see point 4 of the SA algorithm). Indeed, if the rate of accepted perturbations is less than a threshold, then this means that the system has reached the equilibrium.

5. The Schedule Annealing

After each equilibrium state, the temperature has to be decreased carefully. For that, we chose a geometric series, which respects the progressive decreasing of the temperature.

6. Stop Criterion

The stop criterion of clustering is reached when the rate (as in section 5.4) of accepted transitions falls under a threshold which has been set in our experiments to 1%.

7. Energy Computing

The energy to be minimised is expressed by the perplexity of the language model. In order to make the computations easier and faster, we rewrite the formula of perplexity as a recurrent formula. The classical formula of perplexity is given by:

$$PP^{-n} = \frac{N(w_1)}{N(C_1)} \prod_{i=2}^{n} \frac{N(w_i)}{N(C_i)} \frac{N(C_{i-1}C_i)}{N(C_{i-1})}$$

where $N(x)$ is the number of occurrences of event x.
By developing this formula, we obtain:

$$\left(PP_t\right)^{-n} = \frac{\alpha}{Q_1^t} \prod_{i=2}^{n} Q_i^t$$

$$Q_1^t = \frac{1}{N\left(C_1^t\right)} \quad and \quad Q_1^t = \frac{N\left(C_{i-1}^t C_i^t\right)}{N\left(C_i^t\right) N\left(C_{i-1}^t\right)}$$

where PP_t is the perplexity at time t, with $\alpha = \prod_{i=1}^{n} N\left(w_i\right)$,

At time *t+1*, by dropping α (which does not depend on classes) and by moving the word w_d randomly selected from one class to another, the modified perplexity can be expressed as follows:

$$\left(PP_{t+1}\right)^{-n} = \frac{Q_1^t}{Q_1^{t+1}}\left(PP\right)^{-n}\frac{\prod_{j\in B} Q_j^{t+1}}{\prod_{i\in B'} Q_i^t} \quad \text{With}$$

$$B = \left\{\left(C_{k-1}^{t+1}C_k^{t+1}\right)/ w_d \in C_{k-1}^{t+1} \wedge / \vee\, w_d \in C_k^{t+1}\right\}$$

$$B' = \left\{\left(C_{k-1}^{t}C_k^{t}\right)/ w_d \in C_{k-1}^{t} \wedge / \vee\, w_d \in C_k^{t}\right\}$$

Finally, the recurrent formula of the modified perplexity is expressed as follows:

$$PP_{t+1} = PP_t\left(\frac{Q_1^t}{Q_1^{t+1}}\frac{\prod_{j\in B} Q_j^{t+1}}{\prod_{j\in B}, Q_i^t}\right)^{-\frac{1}{n}}$$

By using this formula and by testing several parameters of this algorithm, we achieved several experiments. Figure 1 shows the evaluation of the automatic classification. In this experiment the number of classes has been respectively set to 200 and 1000. It seems that better results in terms of perplexity are obtained using 1000 classes. Figure 2 shows that the manual classification can be improved by using it as an initial classification (a perplexity of 1620[2]) in the iterative process of clustering. At the end, we obtain a new classification, which corresponds to a better perplexity (1206).

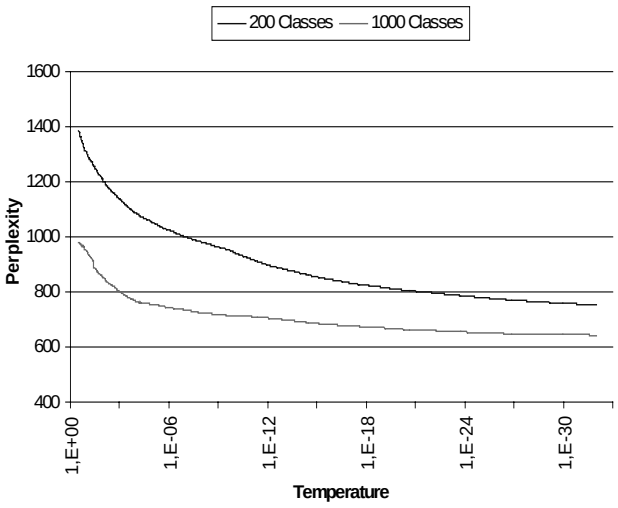

Figure 1: Evolution of the perplexity in accordance with the number of clusters

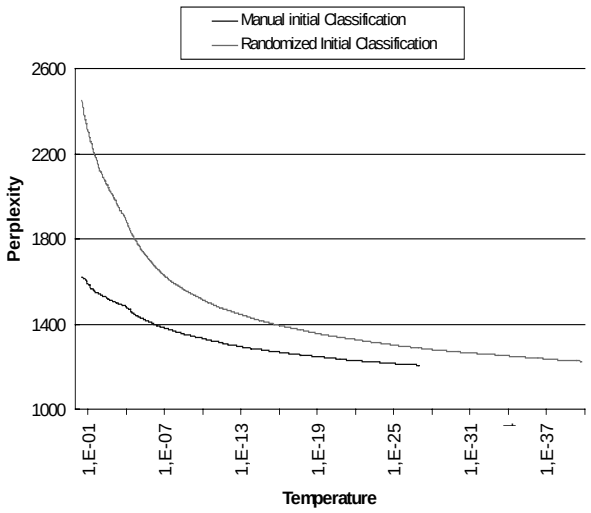

Figure 2: Improvement of the manual classification through the SA algorithm

6. AN OVERVIEW OF THE SHANNON'S GAME

The Shannon game [4] has been adapted in [5] in order to give another method for evaluating language models. A set of truncated sentences is used as a test corpus. The goal of this operation consists in supplying a list of candidate words for each truncated sentence. To each word a bet is associated; which estimates the likelihood of the candidate word. A capital of 1 is distributed between the words of the vocabulary, the perplexity is then evaluated as the inverse of the geometric mean of the bets placed on the correct words. This protocol has been used in a comparative evaluation campaign for language models organized by AUPELF-UREF in which we have taken part [6].

7. DESCRIPTION OF THE LANGUAGE MODEL

Our language model is based on a combination of n-classes and n-grams. In fact, in this model we tried the two kinds of clustering described above.

The language model has been trained on *Le Monde* newspaper (1987-1988), a corpus which contains 42M words. It has also been used to extract the vocabulary (20 000 words) of the speech recognition system.

To explain the different steps of predicting the *n* best words in this experiment, let assume that the truncated sentence is $w_1 \ldots w_{k-1}w_k$, the vocabulary classes is $c_1 \ldots c_K$ and the vocabulary words is $v_1 \ldots v_N$. In the manual clustering experiment, each word v_i belongs to one or several syntactic classes. The prediction is carried out as follows: we label each word w_i of the truncated sentence.

[2] The two experiment have not been achieved on the same corpora.

In order to realise that, we first used our labelling tool, but unfortunately, the results were very bad. This is due to the fact that this tool is efficient only when it disposes of the whole sentence. To deal with this problem, we decided to take into account only the last two words ($w_{k-1}w_k$) of each truncated sentence. That means that all the classes of the words $w_{k-1}w_k$ have been kept, then we assigned the K classes to the word w_{k+1} to be discovered, which can belong to any class. For each class of w_{k-1}, for each class of w_k and for each word w_{k+1}^i of the vocabulary classes (where i refers to the class), we compute the quantity:

$$Q = P(C_{k+1} / C_k C_{k-1}) \times P(w_{k+1}^i / C_{k+1})$$

Finally, the n-class model is used to score again w_{k+1}^i by using a linear combination between the quantity Q and the value of the concerned n-gram.

8. RESULTS

The results described below concern two experiments computed with two sets of clusters of 200 classes. The first set of clusters is the one defined manually and the second one is learned automatically by the SA algorithm. The test corpus is made up of 10 000 truncated sentences, randomly selected from 4 years of *Le Monde Diplomatique*. For each truncated sentence, the language model proposed 5 000 hypotheses which are sorted on the value of their bets. The perplexity is calculated on the whole test corpus (400 000 words). In the manual classification, which we have previously described, a word can be found in several classes, whereas the perplexity presented here can be computed if and only if words belong to one class. Therefore, we have modified the manual classification: all words which belong to several classes have been set in a single class. Then, each of these words has been moved to the class which minimised the perplexity. In table 1, we give the number of the recognised words out of the 10 000 missing, the number of words observed from the first rank to the fifth and the value of the perplexity.

	Manual Clusters	Automatic Clusters
Recognized words	8100	9092
Words at rank 1	1650	1303
Words at rank 1 to 5	3446	4601
Perplexity	1650	1225

Table 1 : Comparative results in terms of observation ranks and perplexity for the 10000 words to be found, including unknown words.

Table 1 shows that better results are obtained with the automatic classification in terms of number of discovered words, words on position one to five and in terms of perplexity. This improvement is not surprising, in fact, the hand classification has been created by following linguistic criteria and the perplexity measure has not been taken into account at all. Whereas the initial goal of the automatic clustering was to improve the perplexity through a long iterative process. That is why, we obtained a better word recognition rate.

9. CONCLUSION

We have presented a comparison between two methods of words classification. The first one is based on a theoretical method inspired from linguistic, and the second one is based on an optimisation technique. Both methods have been evaluated through the Shannon game. In our experiments, we combined manual and automatic classification by improving the first one; through an iterative process. Instead of using a random initial clustering in the simulated annealing algorithm, we utilised the clusters developed by hand. Experiments have shown that the automatic classification outperforms the one based on the linguistic criteria. It gives better results, in terms of perplexity (25% of decreasing) and in terms of recognised words (an improvement of 13% has been achieved). Furthermore, the rate of words recognised in the first five ranks reaches a rate of 46%, whereas with a hand classification, the rate does not exceed 34%. Our future aim is to include this model in our dictation machine (MAUD) and to work on a new method of clustering which reduces not only the perplexity but also the speech word error rate through the iterative process.

10. REFERENCES

[1] M. Jardino, G. Adda « Language Modeling for CSR of large corpus using automatic classification of words » Proc 3rd EUROSPEECH, Vol 2, PP 1191-1194, Berlin, 1993.

[2] L. Moisa, E. Giachin « Automatic Clustering of Words for Probabilistic Language Models » 4th EUROSPEECH, Vol 2, pp 1249-1252, Madrid, 1995.

[3] 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 », Dublin, 1996

[4] C. E. Shannon « Prediction and entropy of printed English » Bell Syst. Techn. J., pp. 50-64, Jan. 1951.

[5] F. Bimbot, M. EL-Bèze, M. Jardino « An Alternative scheme for perplexity estimation », Proc on ICASSP Vol 2 pp. 1483-1486,Munich, 1997.

[6] M. Jardino, F. Bimbot, S. Igounet, K. Smaïli, I. Zitouni, M. El-Bèze « A first evaluation campaign for language models », 1st International Conference on Language Resources & Evaluation, Granada 1998.