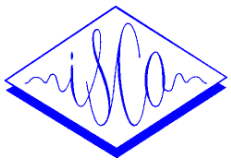

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

A CLASS BIGRAM MODEL FOR VERY LARGE CORPUS

Michèle Jardino

LIMSI-CNRS, B.P.133, F-91403, ORSAY Cedex, FRANCE

ABSTRACT

As pointed out by Jelinek [1] the n-gram word model is a very efficient model but not well adapted for highly inflected languages such as French. So we have developed a class-based bigram model determined entirely automatically from written corpora. The classes are not predefined, the words are not tagged, the solely assumption is the number of classes.
We get a robust model which insures a more complete coverage of the succession probabilities (the studied training text of 2 millions of French words gives a coverage rate of the class bigram model of 50% to be compared with 0.1% for the word bigram model).
Here we present results on new classifications of the text defined above [2], obtained with more than one possible class for each word, as well as optimised combinations of word and class bigram models.

INTRODUCTION

A langage model gives the succession probabilities of words in a text. With stochastic word models these probabilities are easily estimated from countings on training texts. A class-based model requires an efficient mapping of the words into classes. Our model achieves this using an original stochastic method [2,3], which does not need any a priori knowledge such as syntax, grammar or semantics. All we need is a large training text .

BIGRAM CLASS MODEL

AUTOMATIC CLASSIFICATION

Starting with a fixed number of classes, the automatic classification of words is carried out in order to get the highest value of the probability of this training text, in this context. We will show below how this probability depends on the mapping of the words into classes.

Optimisation parameter

The probability of a text of N words: $\omega_1\ \omega_2 \ldots\ \omega_N$, P_T is, ideally:

$$P_T = P(\omega_1)\prod_{n=2}^{N} P(\omega_n \mid \omega_{n-1} \ldots\ \omega_1)$$

In bigram models, the history of each word is reduced to the previous one so that:

$$P_T = P(\omega_1)\prod_{n=2}^{N} P(\omega_n|\omega_{n-1})$$

or, if we gather identical words together and consider word pairs $\omega_L\omega_R$ with ω_L and ω_R as vocabulary words of the training text :

$$P_T = \prod_{L=1}^{V} \prod_{R=1}^{next(L)} P(\omega_R|\omega_L)^{N(\omega_L\omega_R)}$$

where V is the vocabulary size of the text, $N(\omega_L\omega_R)$ the number of bigrams $\omega_L\omega_R$ and next(L) the number of vocabulary words ω_R following each word ω_L.
When words are classified into classes C_k the succession probability of words ω_L and ω_R becomes:

$$P(\omega_R|\omega_L) = \sum_{C_j} P(\omega_R|C_j) \sum_{C_i} P(C_j|C_i)\, P(C_i|\omega_L)$$

We can see from the preceding formulae that different mappings give different values of the probability P_T of the training text: the probability P_T is a characteristic of each classification and will be the parameter to be optimised during the automatic classification. In fact, we use a much more tractable optimisation parameter: the perplexity PP_T [4] which is defined as:

$$PP_T = \exp\left(-\frac{1}{N}\log P_T\right)$$

This expression realises a weighting of P_T by N, the number of words of the text. The smallest values of PP_T corresponds to the best models.
Making the assumption that each word has only one class, it is worthwhile to consider two extreme classifications: first, when the number of classes

equals the number of vocabulary words, it is the classical word bigram model which gives the lowest value of perplexity; then, when words are gathered in one class, this leads to the highest value of perplexity.

Classification process

The training text is automatically classified in order to reach the lowest perplexity value. This is done using a Monte Carlo optimisation method [2].
Probabilities are estimated from maximum likelihood estimations given by counting occurrences of words $N(\omega_R)$, $N(\omega_L)$, of classes $N(C_i)$, $N(C_j)$, and occurrences of successions of classes $N(C_iC_j)$ in the training corpus so that:

$$P_{ML}(\omega_R|C_j) = \frac{N(\omega_R)}{N(C_j)}, \quad P_{ML}(C_j|C_i) = \frac{N(C_iC_j)}{N(C_i)}$$

$P(C_i|\omega_L)$ is the fraction of words ω_L which belongs to the class C_i.
The initial conditions are a fixed number of classes NC, and all words gathered into one class. Then one word randomly chosen is put into a class also randomly chosen; if the new configuration is better the resulting classification is accepted. This process is repeated until a minimum is reached. A global minimum is generally reached, verifications with simulated annealing processes insure this assumption.

CLASSIFICATION RESULTS

Classification have been made on a French corpus made of written texts coming from the newspaper "Le Monde". The training part contains about 2 millions of words for a vocabulary of 65 000 words.
This training set is automatically classified in order to reach its lowest perplexity value.
Using this method we have obtained sensible classification not only in terms of perplexity but also with respect to linguistic considerations (syntactical, grammatical and semantically classes) [5]. This simple model, however, did not take into account the fact that a word can have different classifications depending on the context, for instance, in French "le" can be either an article or a pronoun. We have made a first step to give more flexibility to our model, allowing each word to have at most two classes. Each vocabulary word may be associated to classes depending on the left context: a class containing the subset of words with the highest left bigram, another class with the rest of the words in other left contexts. Other strategies are obviously possible.
Next figure (fig 1) shows values of perplexity during the optimisation process in both cases: "one word-one class", "one word-two classes". The perplexity is reduced by a factor of 25% when two classes may be assigned to each word.

The mapping of the words has interesting features. First, a lot of words don't split into two classes and remain into the same class. Then, very frequent function words like "le", "la", "les" which stand each alone in one class in the former classification, belong now to two classes according to the distinction between article and pronoun. Other ambiguous words are also separated depending on the context: for instance the word "dure" which means "hard" as an adjective and "last" as a verb is effectively mapped into two classes.

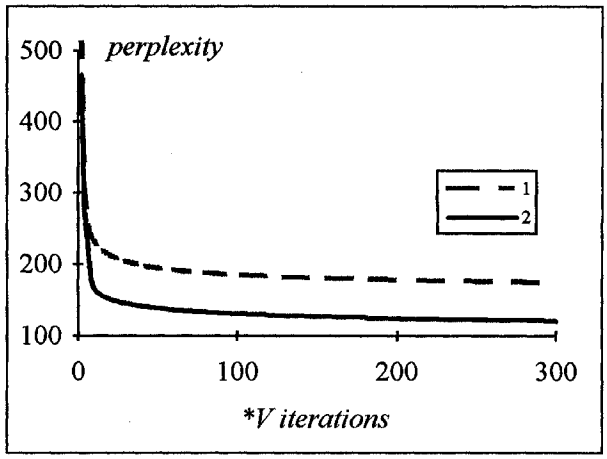

Fig 1: Perplexity during optimisation process
case 1: one class for each word
case 2: two possible classes for each word

BIGRAM CLASS MODEL EVALUATION. COMPARISON WITH BIGRAM WORD MODELS.

In the following, we will consider only classifications obtained with the restriction "one word-one class".
The evaluation of the class-based bigram model is given, as usual, by the perplexity of a test corpus. This test set contains sentences randomly chosen in the original corpus and not contained in the training part. Therefore some events appearing in the test part have not been seen in the training part. In order to evaluate probabilities of these unseen events, different methods of interpolation have been used [6]. Here we describe some of them which we have evaluated on word bigrams and class bigrams.
Two kinds of situations have to be considered in the test set: firstly, the occurrence or not of a word, secondly the occurrence or not of a succession of two words. As pointed out by J. Ueberla [7], we have to

be careful when comparing different models: different processing of unknown words may result in very different values of perplexity. Therefore, unknown words will be processed in the same way for the evaluation and the comparison of bigram-class models and bigram-word models.

UNKNOWN WORDS

Known words (with all inflected forms) come from the training part, we haven't used any dictionary. We have chosen to built a special "class" C_0 which gathers not only the unknown words but also all words of the training set with small occurrences. It is a way to get non-zero probabilities for unseen words in the Katz' spirit [8] where parts of probabilities of less frequent events are redistributed among unseen events. Furthermore this assumption gives probabilities for bigrams containing unseen words. In this special "class" we assume an uniform distribution for all words. The number of unknown words is fixed arbitrarily to a fraction of words of lower occurrence; with 25%, we get underestimated probabilities for the words of the class C_0 in regard of the analysed test set which contains less than 10% of unseen words. But this approximation leads to a very small incidence on perplexity values (about 1%).

UNKNOWN SUCCESSIONS OF WORDS AND CLASSES

Different interpolation methods have been used for the determination of smoothed class bigram probabilities $P_s(C_j|C_i)$ and smoothed word bigram probabilities $P_s(\omega_R|\omega_L)$. The training set gives only maximum likelihood estimations of these probabilities.
The general method is to reduce the maximum likelihood estimations and to redistribute the corresponding "mass" according to the probability rules. Assuming the item X describes as well a word or a class, we define space(X=ω)=V with V the vocabulary size, and space(X=C)=NC with NC the number of classes. The normalisation condition is:

$$\sum_{R \in space(X)} P_S(X_R|X_L) = 1$$

Linear and non-linear interpolation [5] have been used: the parameters involved are either given with [9] or without held-out methods [6,10], or optimized directly on the test set [2]. Here the redistribution has been done over all probabilities and not only on the probabilities of unseen events. This method is easier to implement [9] and will be faster in a recognition task.

Floor method, linear interpolation

This method, generally used in text compression [11], can be written as:

$$P_S(X_R|X_L) = \frac{N(X_L X_R) + \beta\, q}{N(X_L) + \beta}$$

with $\beta > 0$ and q a less specific distribution, defined later. This method is equivalent to the interpolation formula of Jelinek [6] assuming:

$$\lambda = \frac{\beta}{\beta + N(X_L)}$$

Non-linear interpolation

A fixed quantity δ is subtracted from seen bigrams and the corresponding mass $M_L(\delta) = next(X_L)$ is redistributed among all possible bigrams associated with X_L so that:

$$P_S(X_R|X_L) = \frac{Max[N(X_L X_R) - \delta, 0] + M_L(\delta) q}{N(X_L)}$$

δ is optimised directly on the test set but it is possible to determine its value from the training text [6].

Distribution q

The less specific distribution q has to satisfy:

$$\sum_{space(X)} q = 1$$

The following distributions have been experimented:
-uniform on bigrams associated with X_L:

$$q = \frac{next(X_L)}{space(X)}$$

-with backing-off [8], depending on X_R:

$$q = \frac{N(X_R)}{N}$$

The results calculated on a test set of 200 000 words, are given in table 1, assuming for MLU a linear interpolation with a uniform redistribution , for MLBO a linear interpolation with a back-off redistribution , for MNLU a non-linear interpolation with a uniform redistribution and finally for MNLBO a non-linear interpolation with a back-off redistribution.

interpolation	word bigram model	class bigram model	% of words which perform better with bigram-class model
MLU	294	241	75%
MLBO	232	241	58%
MNLU	287	231	75%
MNLBO	228	232	36%

Table 1: Test perplexity for different smoothings

There are less discrepancies between the class bigram models than between word bigram models, which show the robustness of the class-based models. The back-off smoothing technique improves greatly the word bigram model but has no effect on the bigram class model. A more detailed analysis shows that different numbers of words perform better with a model than with the other one, so an appropriate combination of the two models should lead to a better model. This is shown on the figure 2 where new perplexity values are drawn against α, the interpolation parameter, an improvement of about 10% has been obtained at the optimum value of the interpolation parameter.

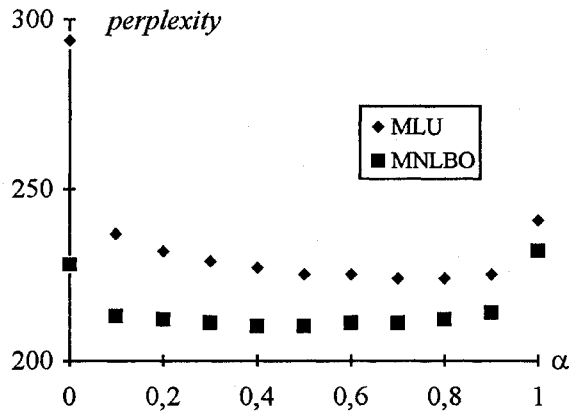

Fig 2: Perplexity obtained with a combination of class and word models for two smoothings.
(α=0 corresponds to the word bigram model, α=1 to the class bigram model)

CONCLUSION

We have built a class bigram model derived from an original classification process. This method is very promising as it does not need neither a priori determination of the classes nor a priori knowledge about the word classes. Words do not need any preliminary tagging. Starting with a fixed number of classes, the criterium of the classification is to minimize the perplexity of the training corpus. We have used this method for a class bigram model with the possibility for a word to have two classes. This model can be extended as well to a class bigram model with more than two classes for a word, as to a class trigram model.
This bigram class model is robust as has been shown by the small influence of the different smoothing techniques experimented. It compares favourably with a bigram word model, at least with a training corpus size of 2 millons of words. A combination of the two models results in a better model with an improvement of about 10% for the test set perplexity.

REFERENCES

[1] F. Jelinek, "Up from trigrams! The struggle for improved language models", EUROSPEECH'91 Proc., Genova, Italy, 1991, p.1037
[2] M. Jardino, G. Adda, "Language modelling for CSR of large corpus using automatic classification of words", EUROSPEECH'93, Proc., Berlin, Germany, 1993, p.1191
[3] M. Jardino, G. Adda, "Automatic word classification using simulated annealing", ICASSP-93, Proc., Minneapolis, USA, 1993, p.II 41
[4] M. Jardino, G. Adda, "Automatic determination of a stochastic bi-gram class language model", ICGI-94, Proc., Alicante, Spain, 1994, to be published.
[5] F. Jelinek, R.L. Mercer, L.R. Bahl, "The development of an experimental discrete dictation recognizer", IEEE, vol.73, n°11,p.1616, Nov.1985
[6] H. Ney, U. Essen, R. Kneser, " On structuring probabilistic dependences in stochastic language modelling", Computer Speech and Language (1994) 8, 1-38
[7] J. Ueberla, " Analysing a simple language model-some general conclusions for language models for speech recognition", Compter Speech and Language 1994) 8, 153-176
[8] S.M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-35, n° 3, March 1987
[9] P. Placeway, R. Schwartz, P. Fung, L. Nguyen, "The estimation of powerful language models from small and large corpora", ICASSP-93 Proc., Minneapolis, USA, 1993, pII-33
[10] K.W.Church, W.A. Gale, "A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams", Computer Speech and Language,(1991) 5, 19-5
[11] I. H. Witten, T. C. Bell, "The zero-frequency problem: estimating the probabilities of novel events in adaptive compression", IEEE Transactions on Information Theory, vol. 37, n°4, July 1991