Towards a Better Collaboration Between a n-class and a n-gram Language Model

K. Smaïli, I. Zitouni, J.P. Haton
LORIA BP 239 54506 Vandoeuvre Les-Nancy
Tel: 03-83-59-20-83 / Fax: 03-83-41-30-79
e-mail: {smaili, zitouni, jph}@loria.fr

Abstract

This paper deals with the combination of a trigram and a triclass. This combination is done in an original manner which is different from a simple linear interpolation. The method, we propose is based on an oriented search for the potential words in a speech process which is simulated through the Shannon game. The combined model presented in this paper outperforms in terms of perplexity, rank of words (for the 100 first propositions) the trigram model and also the one we proposed in the AUPELF (linear interpolation between trigram and triclass). The oriented model reveals that the cooperation between a trigram and a triclass is better when it is used out of the classical linear interpolation.

1. Introduction

The language model is still today one of the most important challenge in automatic speech recognition. Indeed, the speech community made an important effort in order to improve the acoustic rate of recognition. The problem which still today entire is how to overcome the complexity of the natural language through the probabilistic models?
The most popular models are based on n-grams but unfortunately, they are so limited on the modelisation of natural language. One way to improve that is to take into account not only the n-gram but also the n-class structure of the language. The classical way to use both models is by combining them linearly or by using their constraints inside the structure of the maximum entropy [1].
In this paper we present a language model which is based on a combination of a n-gram and a n-class models. Actually, the originality of this approach is in the use of the n-class model, it is not entirely probabilistic. The n-class model is used only to look for the classes in which the likely candidates words are. When the classes are found, the language model bets on each word of each candidate class by using an interpolated trigram.
In the following, we will present the necessity of tagging in order to collect n-class statistics. After a brief review of the principle of Shannon game, we will describe the both methods which have been tested: the linear combination between n-class and trigram and the oriented search models. At the end, experiments about these two models will be described and compared to a classical trigram.

2. The necessity of tagging

The formulation of learning a language model based on classes can be done as follow: given a sentence $W(w_1w_2...w_n)$ how to determine the syntactic categories $C(c_1c_2...c_n)$ that maximises

$$P(c_1\cdots c_n / w_1\cdots w_n)=\frac{P(c_1\cdots c_n)P(w_1\cdots w_n / c_1\cdots c_n)}{P(w_1\cdots w_n)} \quad (1)$$

As we are interested in finding $c_1c_2...c_n$ the common denominator will not affect the computation. By making some independence assumptions, this formula can be expressed as

$$P(c_1c_2\cdots c_n / w_1w_2\cdots w_n)=\prod_{i=1}^{n}P(c_i / c_{i-2}c_{i-1})P(w_i / c_i) \quad (2)$$

Given the formula (2), we can easily understand that text or speech has to be labelled syntactically. In order to estimate the probability $P(c_i / c_{i-2}c_{i-1})$, we need to tag each word of the training corpus. Consequently the dictionary of the application need a syntactic field for each entry. This involves that several words have to be duplicate if they appear in more than one class. From the eight elementary grammatical classes of French, we build up about 233 classes including punctuation. Each punctuation symbol is in a single class. This number of classes was found experimentally to be a good compromise between prediction ability and selectivity. These classes are divided into two groups: the opened and closed classes. A closed class is made up of a finite number of words (such as articles, preposition, ...). An opened class is made up of words which can be formed from root's word (such as verbs, nouns, ...).

In our case a word can belong to until four classes, therefore the formula (1) is correct only if each word is labelled by only one tag. That is why all the corpora we have are labelled contextually. It means, for each word in a context, we give its most likelihood tag. For instance in the sentence "La porte de mon garage". The first two words belong respectively to the classes: {*defined article*, *pronoun*} and {*verb*, *noun*}. But in this context *La* and *porte* will be tagged respectively *defined article* and *noun*.

The probability $P(c_i / c_{i-2}c_{i-1})$ can be expressed as a relative frequency

$$P(c_i / c_{i-2}c_{i-1}) = \frac{n(c_{i-2}c_{i-1}c_i)}{n(c_{i-2}c_{i-1})} \quad (3)$$

Where $n(x)$ counts the number of times that the syntactic structure x occurs in a training text. Actually, the formula (3) is interpolated by 2-class and 1-class. So one of the first steps to do is to collect the counts of 3-class (a sequence of 3 classes), 2-class (a sequence of 2 classes) and 1-class. For that, we labelled a small text by hand and with the statistics collected, we tagged automatically a text of 0,5 million of words extracted from *L'est républicain* (a regional French newspaper). This tagging has been checked by hand and the automatic labelling errors have been corrected. After, we labelled automatically a corpus of 42 million words which represent 2 years (1987-1988) of *Le Monde* (LeM) newspaper. The automatic labelling has been done by a Viterbi algorithm [2].

3. An overview of the Shannon's game

The Shannon game [3] has been adapted in [4] in order to give another method for evaluating language models. A set of truncated sentences is used as a test corpus. The goal of the operation consists of 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[1] (by an agency in charge of the promotion of the French language) in which we have taken part [5].

4. Tests conditions

Three French laboratories take part to this official campaign, all the participants have evaluated their model through the Shannon game and estimated their models on Shannon perplexity, perplexity in situ and in terms of recognized word ranks. The experiments concerned a set of 10000 truncated sentences chosen randomly from a test corpus. This one is made up of 6 years of the French newspaper *Le Monde Diplomatique*. The language model of each participant had to give the 5000 best candidates with their respective bets. This set of hypothesis includes the unknown word (UNK) which is different from the outside list word. In fact, the outside list word is a word which belongs to the vocabulary but the model has not proposed it as a candidate word. In this case, the model set its bet to a uniform probability which depends on the sum of the bets assigned to the 5000 words [5]. To compare the Shannon perplexity to the real perplexity, we computed it on a test corpora made up of the complete 10000 sentences.

5. Description of the language models

Our language model is based on a stochastic model based on the combination of a n-class and a n-gram. This model could use also a unification grammar (UG) for refining the hypothesis but in this version it has not been included because the UG depends on the labeling of the sentence. Unfortunately, our automatic labeler doesn't work correctly with truncated sentences. Actually, in the evaluation, we tested few combination of models.

5.1 A linear combination between trigram and triclass model (LinTrigTric)

It is based on a linear combination of a 3-class and a 3-gram language model which works as follow:

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

1. For the two words which precede the word to discover, the model affects all the possible classes to which they belong. We note these classes respectively Cw_{k-2} and Cw_{k-1} where:
   $w_{k-2} \in Cw_{k-2} = \{ Ci / P(w_{k-2}/Ci) \neq 0 \ \ 1 \leq i \leq 233 \}$ and $w_{k-1} \in Cw_{k-1} = \{ Ci / P(w_{k-1}/Ci) \neq 0 \ \ 1 \leq i \leq 233 \}$
2. In a first version we take into account all the historic of the truncated sentence by labeling it automatically but unfortunately the labeler gave bad results on truncated sentences. That is why, we decided to make an exhaustive search in the classes of the two words which precede the truncated word.
3. Because the truncated word is unknown, the model assigns the 233 classes to this word:

$$v_k \in C_1^{233} = \{C_i / 1 \leq i \leq 233\}$$

At this step a lattice of classes is built up by taking into account all the possible combinations of the Cartesian product .

4. For each path of this classes lattice, we compute Pj :

$$P_j = P(C_j / C^i_{wk-2} C^l_{wk-1}) = \frac{N(C^i_{wk-2} C^l_{wk-1} C_j)}{N(C^i_{wk-2} C^l_{wk-1})} \quad with$$

$$C_j \in C_1^{233}, C^l_{wk-1} \in Cw_{k-1} \text{ and } C^i_{wk-2} \in Cw_{k-2}$$

5. Then for each path and for each candidate word in C_j, the probability of the trigram is combined linearly with the class one and the 5000 best words are kept as potential candidate words. The results of this model have been presented on [5].

5.2 An oriented search model

The precedent model is interesting but the results were not as high as we hoped in terms of perplexity and in number of recognized words. Hence, we decided to use the same model but instead of using a probabilistic model (section 5.1) in all the process of decoding , we used the n-class model only to look for the classes in which the likely candidate words are. In fact, the model acts as in the precedent subsection until point 4. However, the class probability is used only to sort the candidate classes Cj. Then only the *n* best classes are selected to constitute the classes on which we look for candidate words. Each word's bet is computed only by using a trigram model. Two experiments have been done by including and excluding unknown words. The results are summarized on table 1 in which the column (+/-) indicates the difference between the oriented search model in comparison to the precedent one.

	Including UNK	**+/- (%)**	**Excluding UNK**	**+/- (%)**
Number of unknown words	1303	0	1303	0
Number of recognized words	9649	+14,6	8346	+16
Rank 1	2105	+11	1258	+1,9
Rank 1-5	4336	+7	3095	+6
Rank 1-100	7861	+16	6558	+19,5
Rank 1-1000	9219	+16,5	7916	+19
Rank >1000	430	-20	430	-20
Mean Rank	160	-23	185	-25,4
Shannon Perplexity	119,5	-91	179,7	-93

Table 1: Comparative results between LinTrigTric and the oriented search models in terms of recognized words, ranks and Shannon perplexity

Table 1 shows that the new model is better than the one we used in the evaluation campaign for each criterion of this table. One can notice that 78% of the recognized words are in the 100 first propositions and there is only 4,3% of words which are recognized in a rank greater than 1000. The number of words recognized after this rank is the same when excluding or including unknown words. This indicates that unknown words are well recognized and are always in the first candidates. The mean rank of recognized words is 160 which could be considered as a good mean. 96% of words have been recognized, but in this rate we count the unknown words which have been proposed by the language model. Moreover, by excluding unknown words, only 83% of words have been recognized.

Because the trigram is the only model which is used to compute the bets assigned to the candidate words, we canceled the influence of the class model in a second experiment (Table 2), in order to compare our model to a simple trigram.

	Including UNK	+/- (%)	Excluding UNK	+/-(%)
Number of unknown words	1303	0	1303	0
Number of recognized words	9802	+1,5	8499	+1,8
Rank 1	1847	-12	1137	-9,6
Rank 1-5	4003	-7,6	2811	-9
Rank 1-100	7856	-0.06	6553	-0.07
Rank 1-1000	9356	+1,46	8053	+1,7
Rank >1000	446	+3,5	446	+3,5
Mean Rank	166	-3,6	191	-3,1
Shannon Perplexity	177,3	+32,6	282,6	+36,4

Table 2: Comparative results between the oriented search and trigram models in terms of recognized words, ranks and Shannon perplexity

This experiment shows that in terms of perplexity, mean rank and for the first 100 candidates the oriented search model outperforms the trigram model. Indeed, the trigram model yields a perplexity which is 32,6 % worse than the oriented one and this in spite of the fact that the trigram model proposes more words.

6. Conclusion

We compared in this paper a linear interpolation trigram and a triclass models with an oriented search model. This last one uses the triclass only to look for the most likely classes in which we extract the best words. This method gives good results: 96% of words have been discovered, 45% of these words have been recognized in the first five ranks and only 4,5% of these words have been recognized in position greater than 1000. The mean rank is 160 and the Shannon perplexity is 119. This model shows us that the cooperation between the n-class and the n-gram is very successful when the n-class leads the n-gram in the choice of the most likely words. The use of the n-class model, in the prediction process and not as an additive coefficient of the linear combination permits to improve our results and outperforms the model we proposed in the first AUPELF evaluation campaign. This model is also better than a trigram except for the number of recognized words. The trigram model outperforms the oriented search model by 1,5% in term of recognized words but its perplexity is 32,5% inferior. With preserving the advantage of the good mean rank of recognized words and the perplexity, our efforts would have been orienting in the aim to improve the number of recognized words. One track would be the augmentation of the number of classes on which we look for the most likely words.

References

[1] A. L. Berger, V. J. Della Pietra and S. A. Della Pietra « A Maximum Entropy Approach to Natural Language Processing, in computational Linguistics, Vol 22, N1, 1996.

[2] K. Smaili, I. Zitouni, F. Charpillet, J.P. Haton «*An Hybrid Language Model for a Continuous Dictation Prototype*», Proceedings of the 5th European Conference on Speech Communication and Technology, Rhodes, 1997.

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

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

[5] M. Jardino, F. Bimbot, S. Igounet, K. Smaïli, I. Zitouni, M. El-Beze «*A First Evaluation Campaign For Language Models*», Proceedings of the 1st International Conference on Language Resources and Evaluation, PP 801-805, Vol 2 Granada, 1998.