Dealing with distant relationships in natural language modelling for automatic speech recognition

David Langlois, Kamel Smaïli, Jean-Paul Haton

To cite this version:

Dealing with distant relationships in natural language modelling for automatic speech recognition

David Langlois, Kamel Smaïli, Jean-Paul Haton
LORIA
Campus Scientique
BP 239
54506 Vandœuvre-Les-Nancy FRANCE
{David.Langlois, Kamel.Smaili, Jean-Paul.Haton}@loria.fr

ABSTRACT

Classical statistical language models, called n-gram models, describe natural language using the probabilistic relationship between a word to predict and the $n-1$ contiguous words preceding it. Obviously, the linguistic relationships present in a sentence are more complex. A first remark is that there exist distant relationships. We present here some recent work on an alternative model to n-gram models, based on the split of the history, dealing with the interpolation between distant bigram models. More precisely, our model is a cheaper alternative to high order n-grams. In conventional n-grams, when n is greater than 3, events are less frequent and statistics are not reliable. To deal with this problem, and to accurately estimate parameters, we combine a smoothed bigram with distant 3-bigram, distant 4-bigram and a cache composed of 100 words. We present new progresses obtained by using a simulated annealing algorithm in order to calculate the best parameters of this linear combination. With a 20K vocabulary and 40 million words for training, our algorithm improved the perplexity by 5.4% in comparison with the BAUM-WELCH algorithm. Moreover, this new model outperforms a smoothed bigram by 6.1% in terms of perplexity.

Keywords: stochastic language modelling, automatic speech recognition, distant bigram, simulated annealing.

1. INTRODUCTION

Automatic speech recognition systems aim at finding the sequence of words which matches the acoustic signal. Current systems rely on a statistical dynamic alignment between the acoustic sequence and the phrase of words hypotheses. The best alignment is chosen according to the probabilistic matching between the two sequences. Decoding a sentence is then generally provided by a stochastic language model.

The role of a stochastic language model is to provide a probability to a candidate word for recognition, following a sequence of words called the history.

In the present work (cf. [7]), our aim is to study the relationship between the word to predict and its history. We defend the idea that this relationship is more complex than the one supported by classical models. In any natural language, a word is not only related to its immediate neighbors but also to distant words.

Therefore, the model we propose deals with this concept of distance. As it is based on a linear interpolation between several models, the method used to find the best interpolation parameters is important in order to get an efficient model. Thus, we present in this article our recent work based on the use of a simulated annealing (SA) algorithm in interpolation. We also present our work about the cache model, another model which is considered as a particular case of distant language model.

The organization of the paper is as follows. We first explain why n-gram models can not describe complex relationships between the past and the word to predict. Then, we formally describe the language model we propose, in order to deal with this complexity. Then, we describe the interpolation method we use i.e. the SA algorithm. We then present the improvement by the use of SA algorithm in the SHANNON game framework. Then, we discuss the contribution of another distant model: the cache model. Last, we describe how useful is a distant bigram model for a n-gram model.

2. DRAWBACKS OF CLASSICAL MODELS IN LANGUAGE MODELLING

When we use the term classical models, we mean n-gram models, but this section can be also applied to all models which predict a word according to an history, such as n-class models.

For a n-gram model, the history h is made up of the $n-1$ preceding words. It assumes that the word to predict depends on all this history as a single block. Language modelling is thus reduced to a relationship between these two components: the word to predict and its entire history.

In fact, this is not absolutely true. To illustrate our view, let take two examples.

First, in the sentence “The book I bought is brown”, “book” is the subject of “is”. There is a syntactic relationship between these two words. The other words between them (“I”, “bought”) do not participate in this relationship.

Second, in the sentence “This changed conviction into certainty”, we guess there is a relationship between the words “conviction” and “certainty”. This is a semantic relationship: the two words cover the same idea of “evaluation of a fact”.

These two examples show that the relationship between the word to predict and its history is a distant relationship. Moreover, the components of this relationship are isolated words of the history.

In Figures 1 and 2, we show respectively the kind of relationship taken into account by a classical language model and the linguistic relationships which in fact could exist between the past and the word to predict.

Figure 1: Relationship taken into account by a classical language model

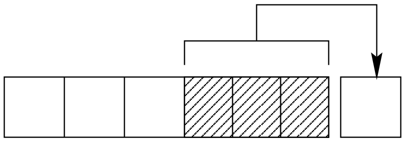
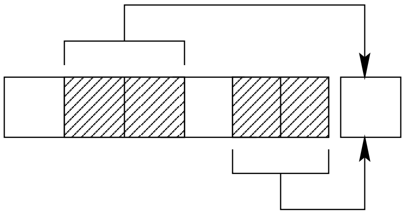


Figure 2: Example of relationships that should be taken into account by a language model



Therefore, to deal with this kind of relationship between isolated words, we propose a model based on distant relationships. In this work, we are interested in isolated words, which is a first step towards a finer modelling. Actually, distant relationships exist also between continuous events (word sequences) and the word to predict. For instance, in “the Empire State Building, this fabulous tower”, there is a strong IS-A relationship between “Empire State Building” and “tower”.

3. THE DISTANT MODEL

The model we propose, inspired from [4], is defined as a linear interpolation between distant bigrams, a bigram, a unigram and a zerogram models. A zerogram model defines a uniform probability over the entire vocabulary. To take into account a history of $n-1$ words, as for a n-gram model, we use n-1 distant bigram models, one for each distant word in the past. By this way, the model takes into account all the history words of the n-gram model. In the following, this model will be called δ-n-gram, where δ is the distant value.

Formal definition

A distant δ-bigram model ($\delta > 2$) in a sentence $w_1w_2w_3\dots w_i$, defines the probability utterance of the word w_i given a word situated $\delta-1$ words in the past. The probability of these-contiguous words is given by:

$$P_\delta(w_i|w_{i-\delta+1}) = \begin{cases} \frac{N_\delta(w_{i-\delta+1},w_i)}{N(w_{i-\delta+1})} & \text{if } N(w_{i-\delta+1}) > 0 \\ \frac{1}{V} & \text{otherwise} \end{cases} \qquad (1)$$

where $N_\delta(v,w)$ is the occurrence of the event w appearing $\delta-1$ words after v, and $N(v)$ is the count of events v. These values are estimated from a training corpus. One can note that a bigram model is a distant bigram model with $\delta = 2$. Nervertheless, in the following we prefer to retain the name of “bigram model” for this one, and to use the notion of distance for $\delta > 2$.

Concerning unseen events (the words not present in the vocabulary), our vocabulary is an open vocabulary. Each word present in a corpus, but not in the vocabulary, is likened to the word UNK (unknown word). UNK is then considered as any other word of the vocabulary.

In order to take into account the whole history, one can hope that the interpolation between $i-1$ distant δ-bigram models will be equivalent to a i-gram model:

$$\begin{aligned} &P(w_i|w_1w_2w_3\dots w_{i-1}) \\ &\qquad = \alpha_0\cdot\frac{1}{V} \\ &\qquad + \alpha_1\cdot P(w_i) \\ &\qquad + \sum_{\delta=2}^{i}\alpha_\delta\cdot P_\delta(w_i|w_{i-\delta+1}) \end{aligned} \qquad (2)$$

where the parameters α_0, α_1 and α_δ vary from 0 to 1 and sum up to 1. These parameters are estimated from a development corpus. $P(w)$ is the unigram probability and V is the size of the vocabulary.

In our experiments, we have limited the maximum value of δ, because our aim was to compare our model with the well known limited history n-gram and n-class models. Therefore the Eq. (2) is approximated by:

$$\begin{aligned} &P(w_i|w_{i-n+1}\dots w_{i-1}) \\ &\qquad = \alpha_0\cdot\frac{1}{V} \\ &\qquad + \alpha_1\cdot P(w_i) \\ &\qquad + \sum_{\delta=2}^{n}\alpha_\delta\cdot P_\delta(w_i|w_{i-\delta+1}) \end{aligned} \qquad (3)$$

The probability of a word sequence is then calculated as it would be done with a n-gram model.

Discussion about parameters

A classical n-gram model requires too many parameters to be trained. As mentioned below, this problem is less important with a distant model.

The history of a distant bigram model is made up of one word. Moreover, this model must give an utterance probability after this history to each word of the vocabulary. In consequence, for a vocabulary size V, the number of parameters is V^2. A δ-n-gram model uses $n-1$ distant bigram models. To this value, we must add the V unigram parameters. Thus, the total number of parameters is $V+(n-1)\cdot V^2$. This value is considerably smaller than the one of a n-gram model (i.e. V^n). So, in theory, the difference reaches 99.99% for a 20K vocabulary. But, in practice many theoretical possible events are never met in the training corpus. To illustrate that, Table 1 gives the effective numbers of distinct events in the training corpus. The above part of the table describes the number of parameters for a trigram model calculated by the CMU TOOLKIT [1]. Such a model deals with trigrams, bigrams and unigrams. The bottom part describes the number of parameters used by a δ-3-gram model. The two models have been trained on the same training corpus.

The real gain is about 52%, which is still good. Note that the difference between the two cells for #bigrams is due to the CMU TOOLKIT which discards events too much rare.

Table 1: Number of distinct parameters for a trigram model and a δ-3-gram model

trigram model	
#trigrams	9187333
#bigrams	1672046
#unigrams	20020
total	10879399
δ-3-gram model	
#distant 3-bigrams	3431974
#bigrams	1704060
#unigrams	20020
total	5156054

Learning parameters by SA algorithm

In order to combine efficiently the different components of the δ-n-gram model, we need to learn the parameters α_i.

Classically, the quality of a model is quantified by its perplexity (PP):

$$\log PP = -\frac{1}{N} \sum_{e \in C} \log P(e) \tag{4}$$

where C is a corpus, of size N in words, P is the distribution of probabilities of the language model over events e.

The perplexity issues from information theory (see [8], [2] or [3]). To simply explain its semantic, PP can be considered as the inverse of the geometric mean of the likelihood of events in C. The smaller this value, the better the stochastic model, since it gives a high value to events in language.

Therefore, finding the best parameters of a δ-n-gram model consists in minimizing an objective function: the perplexity.

There are many methods to find the minimum of a function [9]. In precedent works [7], we used an adaptation of BAUM-WELCH algorithm (see [6] for more explanation).

The approach we use here is based on an SA algorithm [9] which leads to a quasi-optimal global solution.

Let us briefly review the SA algorithm.

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

$$P(E_i \longrightarrow E_j) = \min(1, \exp^{-(E_j - E_i)/T}) \tag{5}$$

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

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

In SA, we have to set the parameters of the algorithm in order to reach the quasi-optimal energy of the system. We have to answer each of the following questions:

- What is the initial state?
- What initial value to choose for T?
- How to decide to lower or not T?
- How to decide that E converges?
- How to perturb the system in order to move to a new state?

In our model, a state is a value for each α_i. The initial state is chosen so as to give the same value to each parameter, without forgetting they must sum up to one. To move to another state, we decrease the value of a parameter, chosen randomly, by a value in $[0 \ldots + \epsilon_T]$. By this operation the parameters no longer sum up to one. Thus, in order to respect this constraint, one increases another parameter, chosen randomly, by the same value. Of course, we prevent changes which lead a parameter out of $]0 \ldots 1[$.

ϵ_T is defined by:

$$\epsilon_T = \frac{\epsilon_0}{N(T) + 1} \tag{6}$$

where $N(T)$ is the number of times T has been changed. The initial value ϵ_0 is set to 0.5. At the beginning, changes could be important. But, due to the decrease of ϵ_T with T, the search of the minimum becomes more and more precise. We guess this phenomenon is natural.

We decide to lower T when the energy stops decreasing. At each lowering, T is divided by 10. The initial value of T is chosen equal to 0.1. The algorithm stops when T reaches a very small value ($1E - 4$), for which we consider that the possibility of change is too small.

Classically, an SA algorithm converges very slowly, due to the very large size of parameters set (problem of the traveling salesman, or word classification [10] for example). But, here, ϵ_T decreases very quickly with T. This fast decreasing could be dangerous with problems with a lot of parameters, since the algorithm requires to much time to find a good minimum of the energy. Our problem, as for it, deals with a small number of parameters: a minimum is thus reached rapidly. We present in Figure 3. the convergence of the algorithm in terms of perplexity for a δ-3-gram. We compared experimentally the SA algorithm with classical minimization methods. We noticed that this method competes very well with other methods, and does not need to use different numerous initialisations to converge to an acceptable value of the minimum.

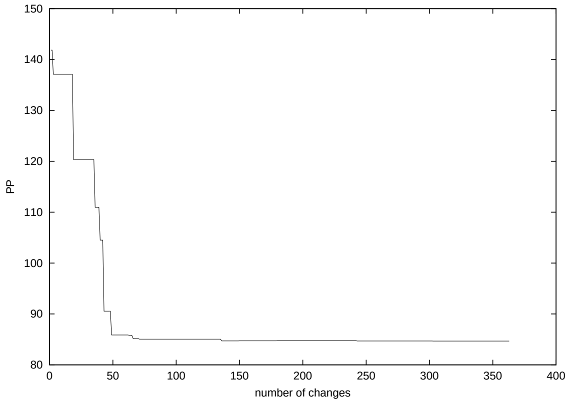


Figure 3: Convergence of the perplexity according to the number of iterations of the simulated annealing algorithm

4. EXPERIMENTAL RESULTS

Several experiments have been conducted on a French corpus. The training corpus is made up of 22 months (about 38M words) of the French newspaper “Le Monde” (1987–1988). Moreover, 2 months have been used to learn the parameters through the SA algorithm. The testing set consists of about 400K words extracted from the French newspaper “Le Monde diplomatique”. δ-3-gram and δ-4-gram language models with a 20K vocabulary have been learned and tested. The out-of-vocabulary word rate of all the language models is 6.5%. Tests were conducted by employing the SHANNON game assessment for the evaluation of language models and by the classical measure i.e. perplexity.

The SHANNON game

This evaluation protocol [5] is derived from the work of SHANNON on the capacity of people to guess missing letters from an unfamiliar text. In [5], the evaluation through the SHANNON game consists in measuring the performance of the model in predicting the word which appears just after a truncated sentence. In our experiments, the test is carried out with 10000 truncated sentences S_i from the newspaper ”Le Monde Diplomatique”. For each truncated sentence S_i, the evaluated model proposes the best 5000 words to appear after it. The model gives a probability to each proposition. The 5000 words scores must sum up to less than 1. Let call Sum_i the sum of the probabilities of the 5000 best words. If the word to be discovered, w_i, is not in the 5000 list, the model assigns a low bet which depends on the scores of the 5000 hypotheses. The value of this bet is the uniform distribution of $1 - Sum_i$ over the remainder of the vocabulary (15000 words in our case). The evaluation step compares each 5000 propositions set with the word which effectively follows the beginning of the sentence. Experimental results appear in Tables 2 and 3 with the following meanings:

- Number UNK: is the number of unknown words over all the 10000 real words to guess,
- Number words in list: is the number of words discovered,
- Mean rank when in list: is the mean rank of w_i calculated over all the truncated sentences, when it is in the 5000 list,
- Number at rank 1: is the number of words observed at the first rank,
- Number at ranks 1 to 5: is the number of words observed in the five first ranks,
- Shannon Perplexity (PP_{Sh}): is an adapted measure of the perplexity. This perplexity is calculated on the values of probability given by the model to all the words to guess,
- Perplexity (PP_{is}): the classical in situ perplexity computed on the corpus from which the truncated sentences have been extracted.

Results: We compared δ-3-gram, δ-4-gram and LMA, a language model based on a combination of a trigram and triclass models. The 3-class model is used only to look for the likely candidate classes. When the classes are found, the language model bets on each word of each candidate class by using an interpolated 3-gram.

The results lead to the following conclusions: models predict more words than LMA. Therefore, our models approach the performance of a more sophisticated language model. Moreover, the performance will certainly be increased by the contribution of other language models based on distance (cache model, triggers . . .). Comparing to the preceding results, the models interpolated by SA are better. For example, the δ-3-gram$_{SA}$ discovered 3620 words at ranks 1 to 5, while this number is 3575 with a δ-3-gram$_{BW}$. Moreover, a δ-4-gram$_{BW}$ is worse than a δ-3-gram$_{BW}$ (except for the words in list), whereas a distant 4-bigram model has to bring more information. With SA, parameters choice leads to a δ-4-gram$_{SA}$ slightly better than a δ-3-gram$_{SA}$. We think the difference for these two models is not important because the linear interpolation tends to mix qualities and drawbacks of all distant bigram models. Last, we notice that, with the SA method, the distance model outperforms the BAUM-WELCH method when excluding unknown words.

Since the difference between perplexities of the models estimated by the same methods are slightly equal, we decided to carry out another test with another corpus of 400K words, as the first one. In this case the perplexity for the model learned with SA decreases by 4.2%.

Combining distant bigrams and a cache model

A cache model [2] models the fact that any word in the past will be probably repeated. This comes from the topic language model which implies to use several times the same words. To estimate this topic probability, a cache model rests on the frequency of words in the large history:

$$P_{\text{cache}}(w|h) = \frac{N(w,h)}{|h|} \tag{7}$$

where h is the history of the word w, $N(w,h)$ is the occurence of w in h, and $|h|$ is the length of h.

Because the cache model is a particular case of distant model, we thus propose to combine a cache of 100 words with a δ-3-gram and a δ-4-gram models. We also used the SA algorithm to calculate the interpolation parameters. Results are presented in Table 4. Each line decribes the linear interpolation between a cache model (with a history made up of 100 words), a distant 4-bigram, a distant 3-bigram, a bigram, a unigram and zerogram models. For each one is given the best linear parameter found by the SA algorithm, and, in the last column, the perplexity calculated on the same corpus test: 400K words from “Le Monde diplomatique”.

It appears that the use of a cache model does not improve highly the perplexity. Nevertheless the combination of the cache model and the two distant bigrams models with the bigram, unigram and zerogram models (perplexity: 92.4) increases the efficiency of the interpolation of this last triplet (perplexity: 98.5). A perplexity reduction of 6.1% is obtained with this model. Unfortunately, this reduction is not very high. This is due to the importance of the bigram in the model. In fact, the bigram contribution is about 85% (see the last line in Table 4), whereas the contribution of all the distant models is only 15%. Even if the cache model has a weak contribution, we think that it is beneficial to include it in order to cope with the phenomenon due to adaptation language models, since it does not require heavy computation nor important memory. In fact, in other language model projects in which we are involved, the distant n-gram and in particular, the cache model are very useful.

Models	LMA	δ-3-gram$_{BW}$	δ-4-gram$_{BW}$	δ-3-gram$_{SA}$	δ-4-gram$_{SA}$
Reference words	10000	10000	10000	10000	10000
PP_{Sh}	119	145.91	145.97	147.35	146.89
Words in list	9649	9835	9847	9841	9850
Mean rank	160	183.91	188.12	184.09	184.28
Words at rank 1	2105	1540	1534	1554	1555
Words at rank 1 to 5	4346	3575	3567	3620	3622
PP_{is}	98	120.92	121.48	119.91	119.17

Table 2: Comparative results for the 10000 words (including unknown words) to be found. Models noted BW are interpolated by the BAUM-WELCH method, SA note models interpolated by simulated annealing

Models	LMA	δ-3-gram$_{BW}$	δ-4-gram$_{BW}$	δ-3-gram$_{SA}$	δ-4-gram$_{SA}$
Reference words	8697	8697	8697	8697	8697
PP_{Sh}	179.74	225.68	225.63	228.27	227.58
Words in list	8346	8532	8544	8538	8547
Mean rank	186	211.74	216.57	211.90	212.09
Words at rank 1	1258	740	719	824	823
Words at rank 1 to 5	3095	2288	2275	2363	2364
PP_{is}	142.99	180.61	181.52	178.76	177.60

Table 3: Comparative results for the 10000 words (excluding unknown words) to be found.

Combining trigram and distant bigram models

It is obvious that a trigram model leads to a better prediction, because the history, as a single block, is more informative. On the other hand, it is difficult to use 4-gram, because of sparse data. It seems that by incorporating a distant 4-bigram model we should obtain a good compromise.

Table 5 illustrates peformances of a simple discounted trigram and a trigram interpolated with a distant 4-gram.

	Interpolation parameters		
Models	Trigram	Distant 4-bigram	PP
Trigram	—		71.61
Distant 4-bigram & trigram	0.97	0.03	70.73

Table 5: Comparative performance between a trigram model and a combination trigram/distant 4-bigram model

Not surprisingly, the combination outperforms the lonely trigram. But as we remarked through the SHANNON game, the performance increase is not very important. We guess that the strength of the word-to-word link decreases very rapidly with the distance. We think that to increase the potential of a distant bigram model its history should be more than one isolated word: a couple of succesive words in the past, for example.

5. CONCLUSION

In this paper, we presented a work on language modelling for speech recognition. In this area, all knowledge sources are probabilistic. That is why the models have to be well learned, and their parameters optimized.

The originality of our model is to take into account the same history as a n-gram model, word-by-word but not as a single block.

In the SHANNON game, distant models interpolated by SA predicted more words in list, at rank 1 and at rank 1 to 5 than the ones interpolated by BW, and more words in list than the LMA model. Moreover, contrary to the models interpolated by the BAUM-WELCH method, the models using SA, and dealing with the distance 4 give better results than the ones dealing with only distance 3. This more logic result shows that the SA algorithm is a better method to optimize our models.

Besides, we incorporated a cache model in the linear interpolation, since such a model can model other linguistic features away from the word to predict. This cache model increases the efficiency of the distant model by 6.1%.

The combination of a distant 4-bigram model with a trigram model is a good compromise between a trigram and a 4-gram models, even if the improvement is still low. We are convinced that the introduction of distant δ-trigram models will improve significantly the results.

6. REFERENCES

[1] P.-R. Clarkson and R. Rosenfeld. Statistical language modeling using the cmu-cambridge toolkit. In *Proceedings ESCA Eurospeech 1997*, volume 5, pages 2707–2710, Rhodes, Greece, September 1997. ESCA, ESCA.

[2] Renato de Mori and Marcello Federico. Language model adaptation. In K. Ponting, editor, *Computational models of speech pattern processing*, volume 169 of *F*. NATO ASI, 1999.

[3] Marcello Federico and Renato de Mori. *Spoken dialogues with computers*, chapter 7, pages 199–230. Academic Press, 1997.

[4] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld. The sphinx speech recognition system:

Cache length history	Cache	distant 4-bigram	distant 3-bigram	bigram	unigram	zerogram	PP
none	0	0	0	0.95	0.04	0.01	98.5
100	0.02	0	0	0.94	0.03	0.01	97.4
none	0	0	0.13	0.85	0.01	0.01	93.4
100	0.02	0	0.12	0.85	0.005	0.005	92.8
none	0	0.01	0.13	0.85	0.005	0.005	93.2
100	0.02	0.01	0.12	0.84	0.005	0.005	92.4

Table 4: Contribution of a cache model to distant models

an overview. *Computer Speech and Language*, 2:137–148, 1993.

[5] M. Jardino, F. Bimbot, S. Igounet, K. Smaïli, and M. El-Beze. A first evaluation campaign for language models. In *Proceedings of the 1st International Conference on Language Resources and Evaluation*, volume 2, pages 801–805, Granada, Spain, 1998.

[6] F. Jelinek, R. L. Mercer, and S. Roukos. Principles of lexical language modelling for speech recognition. *Advances in Signal Processing*, pages 651–699, 1992.

[7] David Langlois and Kamel Smaïli. A new distance language model for a dictation machine: application to maud. In *Proceedings of the Eurospeech 99*, volume 4, Budapest, Hungary, September 1999.

[8] Hermann Ney. The use of the maximum likelihood criterion in language modeling. In K. Ponting, editor, *Computational models of speech pattern processing*, volume 169 of *F*. NATO ASI, 1999.

[9] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C*, chapter 10, pages 444–455. Cambridge University Press, 2 edition, 1992.

[10] K. Smaïli, A. Brun, I. Zitouni, and J.-P. Haton. Automatic and manual clustering for large vocabulary speech recognition: a comparative study. In *Proceedings of the Eurospeech 99*, volume 4, pages 1795–1798. Eurospeech, September 1999.