A Posteriori Multiple Word-Domain Language Model

Elvira I. Sicilia-Garcia, Ji Ming, F. Jack Smith

School Computer Science
Queen's University of Belfast
Belfast BT7 1NN, Northern Ireland
fj.smith@qub.ac.uk

10.21437/Interspeech.2005-18

Abstract

It is shown that the enormous improvement in the size of disk storage space in recent years can be used to build multiple word-domain statistical language models, one for each significant word of a language. Each of these word-domain language models is a precise domain model for the relevent significant word and when combined appropriately they provide a highly specific domain language model for the language following a cache, even a short cache. A Multiple Word-Domain model based on 20,000 individual word language models has been constructed and tested on a Wall Street Journal Corpus. Improvements in perplexity, between 25% and 68%, over a base-line tri-gram model have been obtained in tests

1 Introduction

A human is able to work out the precise domain of a spoken sentence after hearing only a few words. The clear identification of this domain then makes it possible for a human to anticipate the following words and combination of words and thus recognize speech even in a very noisy environment.. This anticipation still cannot be replicated by Statistical language models.

Statistical language models have been improving slowly over the last 20 years due to added complexity and larger training corpora, and using the greatly improved processing power of digital computers. However, they have not been using effectively the enormous increase in the availability of disk storage space, which makes it possible today for a student to buy a 300 GigaByte disk for games and music. This paper suggests one way in which this huge improvement in technology can be used to bring a significant improvement in language modeling, and take a step towards the building of a fairly precise domain model after only a few words of a text

The multiple word-domain language model is based on a simple idea: it extends the idea of cache models [1] and trigger models [2] by triggering a separate n-gram language model for each content word in a cache and combining them to produce an combined model.

This is done as follows. A training corpus for each significant word is formed from the amalgamation of the text fragments in which that word appears, taken from a large global training corpus. In this paper the text fragments are the sentences in which the significant words occur. Experiments have shown that larger fragments are not needed. A significant word is any word that significantly contributes to the context of the text. We define this as any word which is not a stop word, i.e. not articles, conjunctions or prepositions and not some of the most frequently used words in the language such as "will" and not common adverbs and adjectives such as "now", "very", "some", etc. So we assume that all other words are significant and a corpus is built for each. A statistical language model is then calculated from this corpus, i.e. from all of the sentences containing the word. So it should be able to represent the domain of that word. There are a very large number of individual word language models (we have already generated 20,000 of them). The only requirement is enormous quantities of disk space which are now available even on a PC.

2 The Language Models

It was found in experiments that we needed to combine the global language model with the individual word-domain models to obtain good results. (This may be due to the limited size of the global corpus in our tests. 40 MByte.) So we first build a language model for the whole global corpus. Frequencies of words and phrases are derived from the corpus and the conditional probability of a word given a sequence of preceding words is calculated. The individual conditional probabilities are approximated by the maximum likelihoods:

$$P_{ML}(w_i \mid w_1^{i-1}) = \frac{f(w_1^i)}{f(w_1^{i-1})} = \frac{f(w_1 \cdots w_{i-1} w_i)}{f(w_1 \cdots w_{i-1})} \qquad (1)$$

where $f(w_1^n)$ is the frequency of the phrase $w_1^n = w_1 \cdots w_n$ in the text. These probabilities are smoothed by one of a number of well known methods such as Turing-Good estimation [3], the Katz back-off method [4] or others. Although any of these could be used in our experiment to demonstrate the principle of our multiple word-domain model it was convenient to use the empirical weighted average (WA) linear interpolation n-gram

model [5] because of its simplicity. It gives results comparable to the Katz back-off method but is much quicker to use. The weighted average probability of a word w given the preceding words $w_1 \cdots w_n$ is:

$$P_{WA}(w \mid w_1^m) = \frac{\mu_0 P_{ML}(w) + \sum_{i=1}^{m} \mu_i P_{ML}(w \mid w_{m+1-i}^m)}{\sum_{i=0}^{m} \mu_i} \quad (2)$$

where the weighted functions (in the simplest case) are given by:

$$\mu_0 = Ln(T) \quad and \quad \mu_i = Ln(f(w_{m+1-i}^m)) \cdot 2^i \quad (3)$$

T is the number of tokens in the corpus and $f(w_{m+1-i}^m)$ is the frequency of the sentence $w_{m+1-i} \cdots w_m$ in the text.

The unigram maximum likelihood probability of a word is:

$$P_{ML}(w) = \frac{f(w)}{T} \quad (4)$$

The language model defined by equation (2) and (4) is called the global language model when trained on the global corpus. Following the creation of the global model comes the creation of a language model for each significant word, which is formed in the same manner as the global language model.

3 Probability Models

We need to combine the probabilities obtained from each word domain language model and from the global language model, in order to obtain a combined probability for a word given a sequence of words. One simple way to doing this is an arithmetic combination of the global language model and the word language models in a linear interpolated expression as follows:

$$P(w \mid w_1^n) = \lambda_G P_{Global}(w \mid w_1^n) + \sum_{i=1}^{m} \lambda_i P_i(w \mid w_1^n) \quad (5)$$

where $\lambda_G + \sum_{i=1}^{m} \lambda_i = 1$ and $P_i(w \mid w_1^n)$ is the conditional probability in the word language model for the significant word w_i, λ_i is the correspondent weight and m is the number of word models that are included.

Ideally the λ_i parameters would be optimised using a held-out training corpus; however this is not practical as we do not know which combination of words w_i will arise in the cache. So a simpler approach is needed.

3.1 Linear Interpolation

A simple way of choosing the λ values is to give the same weight to all the word language models but a different weight to the global language model, and put a restriction on the number of word language models to be included. This weighted model is defined as:

$$P(w \mid w_1^n) = \lambda \cdot P_{Global}(w \mid w_1^n) + \frac{(1-\lambda)}{m}\left[\sum_{i=1}^{m} P_i(w \mid w_1^n)\right] \quad (6)$$

and λ and m are parameters which are chosen to optimise the model.

3.2 Weighted Probability Model

The weighted probability model is based on the idea that the weight given to a word language model should depend on the size of the training corpora. It is described in the following equation:

$$P(w \mid w_1^n) = \frac{\beta_{Global} \cdot P_{Global}(w \mid w_1^n) + \sum_{i=1}^{m} \beta_i P_i(w \mid w_1^n)}{\beta_{Global} + \sum_{i=1}^{m} \beta_i} \quad (7)$$

where β_{Global} is the weight for the global language model and β_i is the weight for the word model for the word w_i. We give more weight to those word models with small training corpus, as they represent models for the less frequent words which therefore have the most information. The weights used are functions of the size of the word training corpora, that is, of the number of tokens of the training corpora T_i. The optimum functions were found by experimenting with the weights in Table 1.

Table 1: Some of the weights used in this model

Weights
$\mathrm{Ln}(1+\mathrm{Ln}T_i)$
$\mathrm{Sqrt}(\mathrm{Ln}T_i)$
$\mathrm{Ln}T_i$
$\mathrm{Sqrt}(T_i)$
$T_i/\mathrm{Ln}T_i$
T_i
$T_i\mathrm{Ln}T_i$

4 Frequency Models

Instead of combining probabilities to obtain a dynamic language model, it is also possible to combine frequencies before calculating probabilities, i.e. a revised maximum likelihood, replacing equation (1), is:

$$P_{ML}(w_i \mid w_1^{i-1}) = \frac{\lambda_G f_G(w_1^i) + \sum_{i=1}^{m} \lambda_i f_i(w_1^i)}{\lambda_G f_G(w_1^{i-1}) + \sum_{i=1}^{m} \lambda_i f_i(w_1^{i-1})} \quad (8)$$

This can then be combined using the WA model in equation (2). This simple method is automatically normalised and it is easy to implement and fast to execute. The choice of λ is still critical but cannot be optimized from a held out corpus for the same reason that the λ's in equation (5) cannot be optimized: we do not know beforehand which combination of words will occur in the cache. For the frequency model we also combine the frequencies using the same methods that are used for probabilities.

5 Methods of testing

Perplexity is a well known measure of the performance of a language model [6]. We calculate the perplexity of each sentence, w_1^n, using the formula:

$$P(w_1^n) = P(w_1)P(w_2 \mid w_1)\cdots P(w_n \mid w_1^{n-1}) \tag{9}$$

and the perplexity by

$$PP(w_1^n) = \exp\left(-\frac{1}{m}\sum_{i=1}^{m} Ln(P(w_i \mid w_1 w_2 \ldots w_{i-1})\right) \tag{10}$$

There are two methods of calculating the constituent probabilities on the right hand side of equation (10) using the word domain language models, one *a priori* (as reported in earlier papers on this method [7, 8]) and the second *a posteriori*.

5.1 *A Priori* Method

In the *a priori* method at the beginning of the sentence, since we do not know which significant words are going to appear in the sentence, we use the global language model and possibly individual word models from earlier sentences (i.e. from the cache). We then add in a word language model for each significant word after it appears in the sentence. Thus in the sentence

"The cat sat on the mat"

neglecting previous sentences, the first two words are modeled using the global language model, the probability P(sat| the cat) is calculated using the global model combined with the word for "cat", and the last three words are modeled using the global model combined with the word models for "cat" and "sat".

5.2 *A Posteriori* Method

However this is not the only way in which models are tested. We found that for example in domain language models, the researchers extract a whole sentence, paragraph or document from the test file, find all the significant words within it and then they use all of these words to perform an optimisation of the possible domains to find the domain or combination of domains to minimize the perplexity [9,10,11].

To make comparisons with these other domain methods we have also used this *a posteriori* method to calculate perplexity in our model. Firstly all significant words in a sentence are extracted, then a language model is built based on the global model and the word domain models for the significant words. This is then used to calculate the perplexity of the whole sentence. In the example above, "The cat sat on the mat", the perplexity is calculated using the global model combined with the word domain models for the three words "cat", "sat" and "mat" for the whole sentence.

6 Corpus

The methods described above were compared in some experiments using the Wall Street Journal (WSJ) corpus [12]. To compare how the models depend on the size of the training corpus, we test the models in two subsets of the WSJ of 16 million (1988) and 6 million words (1989) approximately. The well known WSJ test file [12] contains 584 paragraphs, 1869 sentences, 34781 tokens and 3677 words types. The results reveal a lower perplexity for the larger 16 million word corpus (as we would expect). So only these results are shown in this paper.

7 Results

The results are shown for the probability model in Tables 2 and for the frequency model in Table 3.

One of the important characteristics of the frequency models is that the probabilities are normalized. This makes this model very quick to calculate as well as being accurate. So this is the version we recommend.

For these models, the number of word models in the cache to reach the maximum performance is 16 and 27 words for the probabilistic and frequency models respectively. So the multiple word-domain language model reduces the size of the cache needed from 500 words as in other models [13, 14] to less than 30 words, which is important for spoken language and closer to the ability of humans.

8 Conclusions

We have shown that the improvement obtained with the *a posteriori* method is well above that obtained for other more computationally expensive domain methods. We accept that *a posteriori* testing gives an advantage to the multiple word-domain model; nevertheless they may be more appropriate for word error rate tests. For perplexity, we feel that the *a priori* results are more probably accurate. However, for humans both are important.

Humans probably hear the sounds of several words spoken before using a form of human language model to make a sensible sentence from the sounds, particularly when there are corruptions. So the idea of using all the words in a sentence to define the domain might be more appropriate than the *a priori* method.

Although we accept that 69% improvement obtained in the *a posteriori* test exaggerates the performance of the multiple word-domain model, we still believe that the use of multiple word-domains, which needs only large amounts of cheap disk space, more precisely models the domain environment of any piece of written or spoken text more accurately than any other domain method. Word error rate measurements are needed to confirm this.

Table 2: Improvement in *a priori* and *a posteriori* perplexity for different probabilistic models with respect to a basic 3-gram model

	Models	**3-gram**	**5-gram**	**Best Values**
	Global Model	0%	19%	
	Linear interpolation Probability Model	9%	24%	λ=0.6 WM=27
A Priori	Weighted Probability Model	11%	25%	Sqrt(T_i) WM=23
A Posteriori	Weighted Probability Model	64%	68%	1/Ln(T_i)

Table 3: Improvement in *a priori* and *a posteriori* perplexity for different frequency models with respect to a basic 3-gram model

	Models	**3-gram**	**5-gram**	**Best Values**
	Global Model	0%	19%	
	Linear Interpolation Frequency Model	8%	21%	λ=0.05 WM=16
A Priori	Weighted Frequency Model	24%	30%	Ln(T_i)/T_i WM=28
A Posteriori	Weighted Frequency Model	67%	69%	1/T_iLn(T_i)

*calculations are in progress.

Note that the change from 26% to 69% is not an improved performance, but rather two different measures of the performance, the second more appropriate for comparison with some other domain models.

9 References

[1] Kuhn R. and De Mori R. "*A Cache-Based Natural Language Model for Speech Recognition*". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 12 (6), pp. 570-583. 1990.

[2] Lau R., Rosenfeld R., Roukos S. "*Trigger-based Language models: A Maximum entropy approach*". IEEE ICASSP 93 Vol.2, pp. 45-48, Minneapolis, MN, U.S.A. 1993.

[3] Good I. J. "The Population Frequencies of Species and the Estimation of Population Parameters". Biometrika, Vol. 40, pp. 237-254. 1953.

[4] Katz S. M. "*Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser*". IEEE Transactions On Acoustic Speech and Signal Processing. Vol. 35 (3), pp. 400-401. 1987.

[5] O'Boyle P., Owens M. and Smith F. J. *"Average n-gram Model of Natural Language"*. Computer Speech and Language. Vol. 8 pp. 337-349. 1994.

[6] Jelinek, F. Mercer, R. L., Bahl, L. R. "*A Maximum Likelihood Approach to Continuous Speech Recognition*". IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 5, pp. 179-190. 1983.

[7] Sicilia-Garcia, E. I., Ming, J., Smith, F. J. "*Triggering Individual Word Domains in n-gram Language Models*" Eurospeech2001, Vol. 1, pp 701-704. Aalborg, Denmark, September. 2001

[8] Sicilia-Garcia, E. I., Ming, J., Smith, F. J. "*Individual Word Language Models and the Frequency Approach*". ICSLP'2002, pp. 897-900, Denver, Colorado, Sept 2002

[9] Seymore, K., Chen, S., Rosenfeld, R. "*Nonlinear Interpolation of Topic Models for Language Model Adaptation*". ICSLP'98, Vol. 6, pp. 2503-2506. Sidneyr, Australia, December 1998.

[10] Iyer, R. M., Ostendorf, M. "*Modeling Long Distance Dependence in Language: Topic Mixture Versus Dynamic Cache Models*". IEEE Transactions on Speech and Audio Processing, Vol. 17, No 1, pp. 30-39. 1999.

[11] Donnelly, P. "*A Domain Based Approach to Natural Language Modelling*" Thesis. Queens' University Belfast September 1998.

[12] Paul, D. B. and Baker, J. M. (1992) "*The Design for the Wall Street Journal-based CSR corpus*", Proceeding of ICLSP 92., pp. 899-902, November 1992.

[13] Clarkson, P. R., Robinson, A. J. "*Language Model Adaptation Using Mixtures and an Exponentially Decaying Cache*". IEEE ICASSP 97, Vol 2, pp. 799-802, Munich, Germany. 1997

[14] Donnelly, P. G., Smith, F. J, Sicilia-Garcia, E. I., Ming J. "*Language Modelling With Hierarchical Domains"*. Proceeding of Eurospeech 99. Vol. 4, pp. 1575-1578, Budapest, Hungary. 1999.