LANGUAGE-MODEL OPTIMIZATION BY MAPPING OF CORPORA

Dietrich Klakow

Philips GmbH Forschungslaboratorien
Weißhausstr.2, D-52066 Aachen, Germany, klakow@pfa.research.philips.com

ABSTRACT

It is questionable whether words are really the best basic units for the estimation of stochastic language models - grouping frequent word sequences to phrases can improve language models. More generally, we have investigated various coding schemes for a corpus. In this paper, this applied to optimize the perplexity of n-gram language models. In tests on two large corpora (WSJ and BNA) the bigram perplexity was reduced by up to 29%. Furthermore, this approach allows to tackle the problem of an open vocabulary with no unknown word.

1. INTRODUCTION

When starting to study correlations within a language and trying to capture its structures in a mathematical form it seems natural to ask for the optimal basic units. A natural choice may be words. But are they the best choice? Part of the motivation for this work comes from text-compression [1, 2]. In that field, there exist two basic ideas which will be summarized here in a simplified manner: The first one uses basic units of widely varying frequency. The length of the symbol encoding a basic unit is chosen according to its frequency. The alternative is to choose code symbols of fixed length and to try to find basic units with nearly identical frequency. Ideas related to the first variant of text compression are widely used in language modeling [1]. In particular, many smoothing techniques are common in both areas. In this paper, the second idea will be investigated for usability in automatic speech recognition and methods to construct basic units whose frequencies are more evenly distributed are considered.

This work was triggered by work by Gauvain et al. [3]. In related work [4, 5], joining words to new phrases has been considered but only for small corpora. In [6], a segmentation of a corpus using maximum likelihood methods for incomplete data is performed and again only applied to small corpora. Finally, some of the language model effects of varigrams [7] can also be captured by phrases. From that perspective phrases may be considered a simplified and efficient variant of varigrams.

2. MAPPING A CORPUS

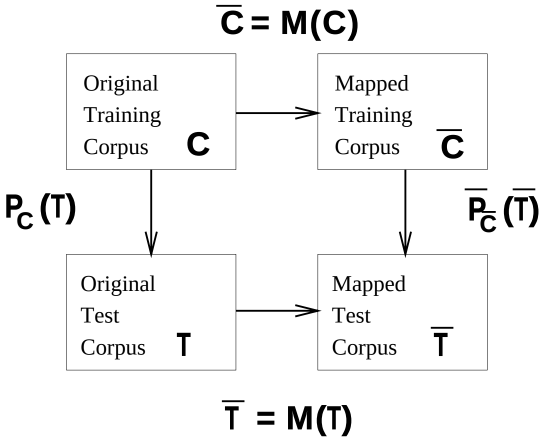
$\bar{C} = M(C)$
Original Training Corpus C
Mapped Training Corpus $\bar{C}$
$P_C(T)$
$\bar{P}_{\bar{C}}(\bar{T})$
Original Test Corpus T
Mapped Test Corpus $\bar{T}$
$\bar{T} = M(T)$

Figure 1: *Mapping of Corpora*

The general idea of this paper is to map a corpus, that is to represent information of the text in a different format. This will result in a modified and hopefully improved vocabulary. It is essential to note that it is not important how the corpus is coded. This is depicted in Fig. 1. The standard way to construct a language model is to estimate probabilities $P_C(T)$ for an utterance T based on a training corpus C. The corpus can also be mapped by a *one-to-one* mapping M. This gives a completely equivalent description of the language model task. Now $\bar{C}$ is used to estimate $\bar{P}_{\bar{C}}(\bar{T})$. It is clear that the test corpus has to be mapped in the same way. Different mappings M_i and M_j on the corpus can be combined to new mappings. But the order of combination matters and the opposite order may give a different result. This causes a technical problem

This work was partially funded by the German Federal Ministry for Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant 01 IV 701 C9. The responsibility for the contents of this study lies with the authors.

for an efficient application of this idea which will be discussed in the next section. Note that text-compression and coding are examples of possible maps M.

3. STANDARD PHRASES

The basic operation of this map which we call "standard phrases" is to join two words (e.g. replacing "White House" by "White_House" or "in the" by "in_the"). The operation of joining two words can be based on various criteria. Three criteria will be investigated:

- the frequency of a pair $N(v, w)$
- the mutual information term $N(vw)\log(\frac{N(vw)N}{N(v)N(w)})$
- the change in the unigram likelihood of the training data as given by

  $\Delta F = \sum_{u\in\bar{V}} \bar{N}(u)\log\frac{\bar{N}(u)}{\bar{N}} - \sum_{u\in V} N(u)\log\frac{N(u)}{N}$

where N is the total number of words, $N(v)$, $N(w)$ and $N(vw)$ the counts of words v, w and the pair vw in the corpus before joining the pair. V is the vocabulary. $\bar{N}$ and $\bar{V}$ are counts and the vocabulary after the mapping.

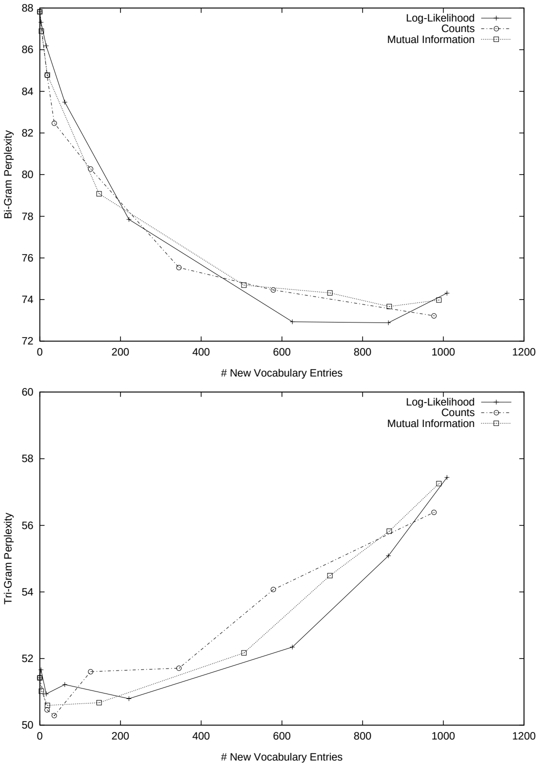

Figure 2: *Results for standard phrases for LDT. Perplexities for bigrams and trigrams.*

The full problem of searching for a vocabulary with evenly distributed frequencies is known to be a NP-complete problem [2]. Hence we will follow a heuristic approach. Ideally, the corpus should be scanned for the best candidate for a transformation and then be transformed accordingly. As this is a very time-consuming procedure, we constructed a sorted list ranked by the particular criterion (i.e. frequency, mutual information, or ΔF). If mutual information or ΔF is the criterion, also lower bound for the frequency of the phrases is set and less frequent phrases are not considered. Within the list all phrases that compete with the highest ranking are pruned. For example, when the best candidate is "in_the" all pairs starting with "the" or ending with "in" are removed from the list. We repeat this with the next best remaining phrase until we obtain a list of non competing phrases. Now, all pairs contained in the list are substituted in the corpus. The whole procedure is repeated until the list generated is empty.

In the first iteration, this procedure will join bigrams to new words. Later steps of the iteration will construct longer phrases like "well_I_think" or "the_White_House". As soon as this causes the frequency of "White_House" to fall below a threshold "White_House" is split into its components again. Such a split-operation is added between two join-operations. The join and split operations are a heuristic approach that tries to minimize the number of iterations.

The results for a small (42 K words) training corpus and a 10 K test corpus for a dictation task from a legal context (LDT) are presented in Fig. 2. The iterations proceed until there are no more candidates for joining or splitting. Different numbers of phrases are obtained by adjusting a threshold for the chosen criterion. In all cases each phrase has to occur at least five times. Note that the perplexities reported are always normalized to the original number of words. For bigrams, perplexity is clearly reduced and at a certain number of phrases the likelihood-criterion may be slightly superior to the other two. For trigrams however, only a minor improvement for a small number of phrases is observed. For more than 200 new vocabulary entries there is a clear increase in perplexity. This may be due to the small size of the training-corpus (overtraining) and is not observed for large corpora as shown in a later section.

4. JOINING NON-ADJACENT WORDS (D1-PHRASES)

A different mapping may be "u v w" → "u_1_w v". The same criteria as in the previous section are used and applied to the pair u w. This will work best when standard phrases are constructed before. Then it is assured that it is better to join the d1-phrase "u_1_w"

n_{std}	$\Delta PP_{std}/n_{std}$	n_{d1}	$\Delta PP_{d1}/n_{d1}$	ΔPP_{total}
16	0.12%	7	0.28%	3.87%
62	0.08%	13	0.13%	6.93%
221	0.06%	40	0.04%	14.57%

Table 1: *Number of standard phrases and d1-phrases (n_{std} and n_{d1}) and improvement per phrase for LDT. The last column shows the total improvement.*

than creating the standard phrases "u_v" or "v_w". However, it is dangerous as some context may be destroyed. Table 1 gives the results for first building standard phrases and then d1-phrases with the ΔF-criterion and the same cut-offs within each line of the table. The table indicates that a bigram can be improved by adding a small number of d1-phrases. For the last line of the table, the standard phrases reduce perplexity by 12.8% that is 0.06% per standard phrase. The d1-phrases give an additional improvement of 1.6% that is 0.04% per d1-phrase. Thus we can conclude that the improvement by d1-phrases is comparable to the standard phrases but there is a smaller number of them. Also, d1-phrases are constructed after standard phrases and hence many good candidates for d1-phrases are already integrated into standard phrases containing more than two words. The best scored d1-phrase is "the_1_of". The most frequent occurrences are "the_1_of subject", "the_1_of motions" and "the_1_of amount". The next two d1-phrases are "to_1_the" and "the_1_of_the". An example from the corpus is "to match the_1_of_the terms purchases".

5. SWAPPING TWO WORDS

To fix some of the problems that might occur with d1-phrases, swapping two words in a given context based on the log-likelihood gain was considered. The elementary map is $w_1w_2w_3w_4 \rightarrow w_1w_3w_2w_4$ and $w_1w_3w_2w_4 \rightarrow w_1w_2w_3w_4$. The indexes refer to a particular set of words chosen. The second part of the map is necessary to ensure a one-to-one mapping. In this mapping the counts of the unigrams are constant and the bigram counts change as follows: $\bar{N}(w_1, w_2) = N(w_1, w_2) - N(w_1, w_2, w_3, w_4) + N(w_1, w_3, w_2, w_4)$. The counts for the five other pairs concerned change accordingly. The change in log-likelihood is easily deduced from the modified counts. A first variant of application considered each 4-tuple of words separately thus really taking into account the context of the two candidate words for swapping. The second variant averaged over w_1 and w_4. Both variants gave negligible improvements in perplexity.

6. SPELL INFREQUENT WORDS

Finally we want to introduce still another mapping: the spell operation. This will be employed after standard phrases are constructed. Words that are not in a list of words are split into letters and afterwards letters will be joined as described in section 3 based on one of the criteria described there. Thus the vocabulary will be a mixture of phrases, words, syllables and letters. Note that such a model will score the *whole* corpus. There are no out-of-vocabulary sections any more!

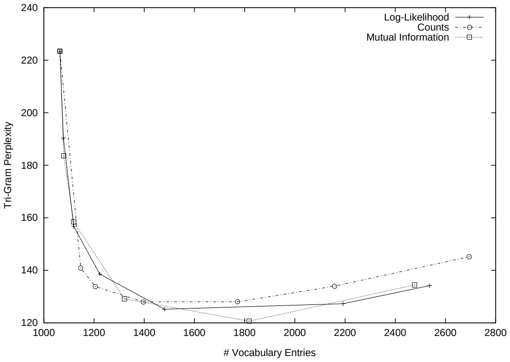

Figure 3: *Trigram perplexity for a vocabulary of words, syllables and letters for LDT.*

The motivation for this is twofold. Firstly, there is the problem of an open vocabulary. We will assume that the acoustic module will hypothesize a text-string which has to be scored. The string may contain words not seen in the training material. Still, instead of falling back to a zero-gram, a better estimate can be obtained based on letter and syllable frequencies. This will yield rather high scores but it may also yield a more discriminative model.

The second motivation comes from language model algorithms. Some algorithms have a complexity scaling of V^N ($N-1$ is the length of the history) and hence a smaller effective size of the vocabulary may be desirable. E.g., for a 64 K recognition vocabulary, only 16 K entries may be kept as words and the remaining words spelled and joined to new units.

It is difficult to compare the proposed language model to established ones. On the parts of the test-set, seen in the training, nothing changes, but the parts that have not been scored before are now also taken into account. It cannot even be compared to models where unseen words are mapped to an OOV-symbol because this dramatically underestimates the perplexity. The ultimate test would be a rescoring experiment which was not performed yet. In Fig. 3 we present the trigram perplexity. It falls significantly as the size of the vocabulary increases. However, we also observe some over-training here, as the vocabulary becomes too large.

7. PHRASES ON WSJ AND BNA

In this last section an application to two large corpora is presented. Only standard phrases were tested. For the Wallstreet-Journal Corpus (WSJ), which consists of about 40 million words, the results are presented in Table 2. The vocabulary size is 5 K words. For 226 phrases the trigram perplexity is reduced by 7.2%. Since the corpus is larger there is no effect of overtraining. For the 4-gram the improvement is still 3.0% and only for the 5-gram there is no change. As for some applications very good performance of bigrams is important we also present results for a large number of phrases. Here the improvement is 29%. This will be helpful in situations where there is not enough memory available for an application to use a trigram. Without phrases the bigram needs 21 MB and the trigram 79 MB. Even the set-up with a large number of phrases needs only 25 MB.

N	0	226	3831 Phrases
1	738.0	562.6	335.1
2	113.0	100.0	80.3
3	60.8	56.4	-
4	52.6	51.0	-
5	50.4	50.3	-

Table 2: *Perplexities for WSJ.*

N	without	with Phrases
1	1026.4	841.2
2	257.1	235.4
3	180.0	172.7

Table 3: *Perplexities for BNA.*

The Broadcast-News-Archive (BNA) is the largest set-up we tested. The training corpus consists of 140 million words of transcribed broadcast-news. The vocabulary size is 64 K words and 330 phrases are constructed (Table 3). The improvement of the bigram is 8.4% and for the trigram 4.1%. Joining phrases increases the effective length of the words. For 226 phrases on WSJ the average length of the new words in terms of the old words is 1.13. When constructing 3831 phrases this value increases to 1.35. For BNA we have 1.16. This shows the limits of the method. The unigram context is still quite short and N-gram models are necessary to model the relation between the new basic units. However, there are also a few examples of very long phrases. For the set-up with 3831 phrases the longest one (with 1117 occurrences in the corpus) is "in_New_York_stock_exchange_composite_trading_yesterday".

8. CONCLUSION

The concept of corpus-mapping is applied to language modeling. We discussed four different basic mappings and studied their effect. Joining words gives large perplexity reductions and spelling infrequent words may be a method to tackle open vocabularies. A small number of d1-phrases helps to capture certain grammatical structures that are otherwise more difficult to model. Tests on WSJ and BNA showed the applicability of the scheme proposed to large-vocabulary standard tasks.

9. ACKNOWLEDGMENT

The author would like to thank Stefan Besling, Peter Beyerlein, Reinhard Kneser, Jochen Peters and Patricia Wilcox for many stimulating discussions.

10. REFERENCES

[1] T.C. Bell, J.G. Cleary, and I.H. Witten: "Text Compression", *Prentice Hall*, 1990.

[2] J.A. Storer: "Data Compression", *Computer Science Press* , 1988.

[3] J.L. Gauvain, G. Adda. L. Lamel, and M. Adda-Decker: "Transcribing Broadcast News: The LIMSI Nov96 Hub4 System", *DARPA Speech Recognition Workshop*, pp. 663, 1997.

[4] K. Ries, F. D. Buo, and A. Waibel: "Class Phrase Models for Language Modeling", *Proc. ICSLP*, pp. 398, 1996.

[5] K. Hwang: "Vocabulary Optimization based on Perplexity", *Proc. ICASSP*, pp. 1419, 1997.

[6] S. Deligne and F. Bimbot: "Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams", *Proc. ICASSP*, pp. 169, 1995.

[7] R. Kneser: "Statistical Language Modeling Using a Variable Context Length", *Proc. ICSLP*, pp. 494, 1996.