PHRASE BIGRAMS FOR CONTINUOUS SPEECH RECOGNITION

Egidio P Giachin

CSELT - Centro Studi e Laboratori Telecomunicazioni
Via Reiss Romoli, 274 - 10148 Torino, Italy

ABSTRACT

In some speech recognition tasks, such as man-machine dialogue systems, the spoken sentences include several recurrent phrases. A bigram language model does not adequately represent these phrases because it underestimates their probability. A better approach consists of modeling phrases as if they were individual dictionary elements. They are inserted as additional entries into the word lexicon, on which bigrams are finally computed. This paper discusses two procedures for automatically determining frequent phrases in an unlabeled training set of written sentences. One procedure is optimal since it minimizes the set perplexity. The other, based on information theoretic criteria, insures that the resulting model has a high statistical robustness. The two procedures are tested on a 762-word spontaneous speech recognition task. They give similar results and provide a moderate improvement over standard bigrams.

1. INTRODUCTION

Spoken dialogue systems that enable a human to access information in a remote database using a telephone have recently been tested with real, unexperienced users by several research labs. The sentences used by real people interacting with such a system, called spontaneous speech, have some peculiarities that differentiate them from those collected in a controlled environment [1]. One typical feature of these sentences is that they contain a consistent number of short phrases that have a very high frequency of occurrence. This paper investigates methods to account for this kind of linguistic structure within the framework of a probabilistic language model.

Bigrams are an effective means for capturing natural language regularities and can be efficiently integrated in a recognizer based on Viterbi decoding. They however are not an excellent model for frequent phrases: bigrams only represent constraints between adjacent words and hence underestimate the phrase probabilities. A simple method to overcome this problem consists of modeling phrases as if they were individual dictionary elements. These elements are inserted as additional entries into the word lexicon. The training text is translated in order to substitute the word phrases with their corresponding symbols, and finally bigrams are computed on the extended lexicon. In this way, the recognizer does not accumulate score when decoding the words inside a phrase, though normally adds a bigram score when a phrase boundary is traversed.

Phrase modeling is advantageous only if it can be performed automatically. In [4] an iterative procedure to automatically find the phrases to model was described. The procedure, called bootstrap, is *optimal* since it is designed to minimize perplexity. (A similar approach was independently proposed in [7].) The present paper completes the work of [4] by discussing results in terms of recognition accuracy. Moreover, it experimentally compares the above procedure with another method to find phrases, called *heuristic*, that attempts at associating words according to their mutual information. (An algorithm of this kind, similar to the one studied here, was first proposed by R.L. Mercer in [5]). Experimental results show that, although the optimal bootstrap procedure does achieve the lowest perplexity, the heuristic method produces values that do not differ much from it. When tested with speech data (1202 spontaneous speech sentences from 34 speakers, PBX quality, 762 words, 310 context dependent units modeled through continuous density HMMs), the two methods give very similar word error rate, and in some cases the heuristic method outperforms the optimal one, probably because the former one is statistically more robust than the latter. These results, while showing that statistical robustness interferes with perplexity in determining the quality of a language model, also suggest that the wide-range linguistic structure of practical utility for a recognizer lies in a small number of frequent phrases; different methods can identify most of them, and the choice of a particular method is unimportant.

2. PHRASE BIGRAMS

The role of the language model in a speech recognition system is that of estimating the prior probability of the word sequences occurring in the task. The bigram model is one of the most successful approaches for that purpose. In tasks for which the training database is small and does not permit the computation of bigrams for each individual word pair, the *class bigram* model is used. This approach better copes with the sparseness of training data: words are clustered into equivalence classes, and the inter-word transition probability is assumed to depend only on the word classes. We will refer to word classes rather than words throughout this paper.

The limit of bigrams is that they only capture relationships between adjacent words; wider-range models are necessary in order to better account for larger scale natural

0-7803-2431-5/95 $4.00

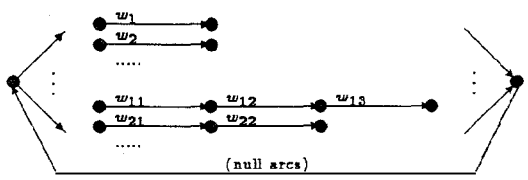

Figure 1: Representation of words (w_i) and phrases (w_{ij})

language structure. This, in turn, demands for larger training databases. For the problem we are addressing, however (database access through a spoken dialogue system [2]), the collection of training corpora is an extremely costly procedure. A typical corpus may have a number of words of the order of 100,000, which makes the training of even trigrams very unreliable. Hence different types of high-order language models have to be sought for. A key to cope with this problem is the fact that, in the above mentioned tasks, the sentences collected from unexperienced users who interact with the system contain a consistent number of repetitive phrases pertaining to the domain (e.g., for the train timetable enquiry task of [2], *"I would like to know"*, *"from Torino"*, *"at two fifteen"*, *"first class fare"*, *"what type of train is it"*, etc.) This suggests to model frequent phrases through arc sequences of a finite state grammar, as in Fig. 1; bigrams are then computed on an extended set of elements that include both single words (which are still necessary to model the less frequent phrases) and word sequences. Thus phrases are treated as though they were single lexicon entries.

To determine which sequences should be represented this way, an optimal procedure exists [4]. The procedure is optimal in that it is bound to reach a minimum of perplexity. We well refer to this procedure as the *optimal* one. A perplexity reduction of about 20% was obtained with respect to standard word bigrams, depending on the lexicon and prior word clustering used, and a 5% reduction with respect to trigrams. In the following we will discuss recognition results, in terms of word accuracy (WA), when the model produced by this procedure is applied during acoustic-phonetic decoding. Moreover, we will compare the optimal procedure to another method, called *heuristic*, that attempts to find phrases by looking at mutual information between words.

3. THE OPTIMAL PROCEDURE

The procedure is entirely automatic and works according to a perplexity minimization criterion. It starts by identifying the two words that, when connected into a single element, produce the highest perplexity reduction. By cyclically repeating this action, it "bootstraps" to longer and longer word chains. This procedure is accomplished without human supervision and does not require any prior manual segmentation or labeling of training data.

The core of the algorithm is the computation of the log-probability of a training text of word classes $\underline{c} = (c_1, \ldots, c_N)$ before and after two adjacent word classes c_a and c_b are connected into a phrase, i.e. they are replaced by a new word class c_{ab} whenever they occur in sequence. This amounts to estimating bigrams on the modified text $\underline{c}'$, resulting from the substitution of every sequence (c_a, c_b) with the symbol c_{ab}, and to compute the probability of the modified text according to the new bigrams. The resulting algorithm, reported from [4] for clarity, is as follows:

1. Begin with a training text of words $\underline{w} = (w_1, \ldots, w_N)$, and the associated word class text $\underline{c} = (c_1, \ldots, c_N)$, resulting from having partitioned words among M possible classes out of a predefined set C; let $P(\underline{c})$ be its probability computed through bigrams on C.
2. Determine the two word classes c_a, c_b that, when connected, give rise to the highest probability of the training text, i.e. the classes corresponding to
$$\underset{c_1 \in C, c_2 \in C}{argmax} \; \hat{P}(\underline{c}, c_1, c_2)$$
where $\hat{P}(\underline{c}, c_1, c_2)$ is the log-probability of the modified training text when classes c_1 and c_2 have been connected, as described above;
3. If $\hat{P}(\underline{c}, c_1, c_2) > P(\underline{c})$, add the new class c_{ab}, deriving from the connection of c_a and c_b, to the set C, and modify the training text accordingly;
4. Loop to point 2 until $\hat{P}(\underline{c}, c_1, c_2)$ does not change from the previous iteration.

This algorithm is bound to converge to a maximum of log-probability (i.e. a minimum of perplexity) on the training text, though the optimum is not insured to be the global one. The problem of convergence on the test set and the effect of introducing the leaving-one-out technique, as well as other possible variants, are discussed in [4].

4. THE HEURISTIC PROCEDURE

A different procedure to find out frequently occurring phrases is based on information theoretic considerations. The idea is that a "frequent" phrase should appear as a stable "island" of words that have a high probability of co-occurring in sequence. That is, the adjacent words that appear in the phrase should be highly correlated between them; conversely, the words that lie at both ends of the phrase should have low correlation with outer words. If one takes mutual information as a measure of correlation between adjacent words, the following iterative procedure can be defined:

1. Begin with a training text $\underline{w} = (w_1, \ldots, w_N)$, and the associated word class text $\underline{c} = (c_1, \ldots, c_N)$, as for the optimal procedure.
2. Determine the two word classes c_a, c_b that correspond to a maximum of mutual information
$$\underset{c_1 \in C, c_2 \in C}{argmax} \; \log \frac{P(c_1, c_2)}{P(c_1)P(c_2)}$$
3. Add the new class c_{ab} to C and modify the text, as for the optimal procedure.
4. Loop to point 2 until a convergence criterion is met (see below).

By itself, this algorithm does not converge. However, if one continues to add new classes (phrases), it is found that the test set perplexity decreases first, but increases

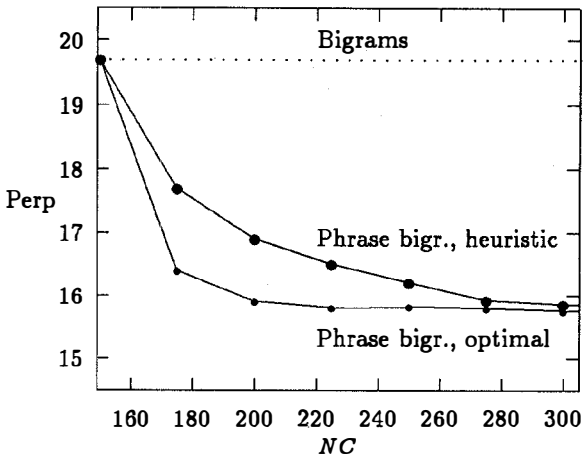

Figure 2: Perplexity vs. no. of classes NC

afterwards. The point in which perplexity begins to increase can then be used as a stopping criterion. The dependency of perplexity on the number of cycles is not regular, however, which makes the determination of the stopping point rather arbitrary. Also, a very high number of new classes (phrases) are usually generated before perplexity begins to increase. A different stopping criterion is then given by the number of produced classes: if it is too high, phrase bigrams will not be trained properly and hence it should be kept from growing excessively.

The algorithm need not necessarily use mutual information: different correlation measures may be employed. Mutual information as defined above proved to provide the best results (though data are similar). Several variants of this algorithm are possible. For example, one could produce the N top ranking couples at each cycle instead of only one, or take all the couples with mutual information above a predefined threshold (the latter approach was cited in [5]). Some variants have been tested; no significant performance differences were observed between them.

This algorithm is not optimal. However, it guarantees that the word associations found are statistically robust. The perplexity values it provides do not differ much from those of the optimal algorithm.

5. EXPERIMENTAL RESULTS

Experiments have been carried out on a task referring to a train timetable database enquiry using a spoken dialogue system with a lexicon of 762 words [2]. A corpus of spontaneous speech was collected from real users who interacted with the system through a telephone connected to the local PBX [1]. The corpus contains 9466 sentences for a total of 60089 words. A small portion of this corpus had previously been held out to perform testing of both acoustic and language models, and therefore did not contribute to training. This portion has been furtherly purged of the sentences (about 10%) that contained out-of-vocabulary words and hence could not have benefited much from the language modeling technique under study. There remained 1202 sentences for testing. Words were clustered into 150 classes using an automatic clustering algorithm that provided better results than those reported in [4]. Recognition was performed using the setup described in [3]. 310 context-dependent subword units are modeled through discrete density HMMs. A Viterbi algorithm is used for decoding; it performs a one-pass search using a stochastic finite-state grammar and is therefore able to support both word and phrase bigrams.

Setup	*Perpl.*	*Word error*	*Sent. error*
Bigram	19.7	17.4%	37.9%
Phr.Big., Optimal	15.7	15.8%	34.7%
Phr.Big., Heuristic	15.8	15.7%	34.2%

Table 1: Recognition results

5.1. Perplexity results

Both procedures are expected to output new classes (phrases) ranked according to their effectiveness. It is therefore interesting to look at the perplexity values obtained for the test set as long as new classes are being produced. Since the test set may include pairs never seen in the training set, linear smoothing has been used to compute bigrams [6]. Fig. 2 shows the test set perplexity measured every 25 new classes (in addition to the original 150 word classes) output by either procedure. In both the optimal and the heuristic algorithm perplexity decreases rapidly with the no. of classes NC, and remains virtually stable thereafter. In the heuristic algorithm it tends to increase, though irregularly, for $NC \simeq 350$. (Also in the optimal procedure the test set perplexity increases slightly, because minimization was done on the training set and no leaving-one-out technique has been used.) The optimal algorithm does find a lower perplexity than the heuristic one, however the two figures do not differ much. A perplexity reduction of about 20% is obtained with respect to standard bigrams. A significant fact shown by Fig. 2 is that most of the perplexity reduction is due to the first 40 or 50 new classes (phrases) found. This result indicates that, for the recognition task under consideration, the wide-range natural language structure that can be of practical utility in a bigram-like approach lies in a small number of frequent phrases.

5.2. Recognition results

A summary of recognition results is presented in Table 1. Results are given in terms of perplexity, word error rate, and sentence error rate. The baseline setup includes standard bigrams computed over the 150 word classes. The two other rows refer to the above algorithms. The optimal one produced 293 classes; the heuristic one was stopped at 300 classes. In all cases bigrams have been computed using linear smoothing.

Data show that, though the heuristic method does not achieve a perplexity value as low as the optimal one, its performance in recognition is about the same, actually slightly better (though differences are not statistically significant). This is probably due to the high statistical robustness of the phrases found by the heuristic method. On the other hand, about 70% of the phrases found are common to both methods. These are the most frequent ones and largely deter-

Set	*Baseline*		*State dep.*	
	Bigram	Phr.Big.	Bigram	Phr.Big.
TP	35.7/20.1	32.0/18.8	13.2/11.0	13.2/11.0
FP	34.9/19.3	32.3/17.1	15.2/ 9.4	15.2/ 8.8
T	26.0/22.7	21.8/21.6	14.4/18.2	13.7/17.0
I	15.4/15.0	11.9/13.9	13.5/14.9	10.9/13.6
F	28.4/21.4	23.2/18.6	24.3/21.0	20.9/18.7

Table 2: Perplexity/Word error rate, baseline and dialogue state dependent training

mine the recognition results; different methods can identify them, and the choice of a particular method is unimportant.

The decrease of error rate provided by any method is rather small. However it is not lower than expected by looking at perplexity alone. An empirical rule found by some researchers has it that the word error rate is roughly proportional to the square root of perplexity. In our case, the perplexity decrease would suggest a 9% decrease of word error rate, not much different from the actually observed figure. This again indicates that most of the constraining power lies in a small set of frequent phrases. Larger-scale linguistic structure is apparently either too difficult to be captured by a finite state formalism, or too rarely occurring in order to be trained properly and give useful constraints during recognition.

5.3. Dialogue state dependent modeling

The spontaneous speech corpus has been collected through a dialogue system that passed through numerous different dialogue states during each conversation. In some states the system asks focused questions and hence constrains what the user is about to say; in others the system leaves free initiative to the user. The usage of state-dependent language models should therefore improve the recognizer performance. Training and test sentences were partitioned into five subsets according to what the user was asked to say:

ToPlace (TP) An arrival city or station.

FromPlace (FP) A departure city or station.

Time (T) A time expression (typically a departure time).

Info (I) Any information on train services, fares, etc.

FirstSentence (F) The very first dialogue utterance (the user is not constrained in any way by the system).

The above subsets are empirically ordered from the most to the least focused. Table 2 compares the word bigram and phrase bigram models as tested for each of the five sets. Training has been done using either the whole sentence set (*Baseline* portion) or the state-dependent subsets (*State dep* portion). Both perplexity and error rate are given for each column. Data for the phrase bigrams refer to the optimal procedure. Data show that state-dependent training improved all models. The improvement is not uniform throughout all sets and models. For the more focused sets (*TP* and *FP*) the phrase bigrams perform better than when they are trained on the whole set; this is however true also for bigrams, which makes the use of phrase bigrams less useful. The reason is that many of these sentences consist of just two words, for which word and phrase bigrams do not behave much differently. Conversely, the advantage over bigrams is higher for the more varied sets (*I* and *F*), though the improvement due to state-dependent training is much smaller. By taking the best models for each subset an overall error rate of 14.8% is obtained for state-dependent phrase bigrams.

5.4. Other heuristic procedures

As was said in Section 4, other variants to the heuristic procedure are possible. Some of them have been tested and did not show any improvement as far as perplexity or recognition error rate are concerned. They also produced the same core group of frequent phrases that mostly contribute to the decrease of perplexity. The remaining phrases, however, differ, and some are similar to linguistically meaningful grammatical constituents. A further study along this direction may prove useful to design methods for automatic partial parsing of large corpora.

6. CONCLUSIONS

Two methods to overcome the limit of bigrams have been investigated, which are less sensitive than trigrams to sparseness of training data. Both methods are based on the use of bigrams of phrases as well as of words, and identify phrases automatically. To this goal, the first method (optimal) directly aims at minimizing perplexity. The second one (heuristic) attempts at finding word associations by looking at their mutual information. Though the optimal method finds the best perplexity value, the heuristic one produces similar values. Recognition data show that the two methods have similar performance, and provide a moderate improvement over word bigrams. Results indicate that most wide-range linguistic constraints lie in a small number of frequent phrases; longer or rarer phrases do not seem to contribute any benefit.

REFERENCES

[1] P. Baggia, E. Gerbino, E. Giachin, and C. Rullent, "Experiences of spontaneous speech interaction with a dialogue system", *CRIM/FORWISS Workshop*, München, September 1994.

[2] D. Clementino and L. Fissore, "A man-machine dialogue system for speech access to train table information", *Eurospeech 93*, Berlin, September 1993.

[3] L. Fissore, E. Giachin, P. Laface, and P. Massafra, "Using grammars in forward and backward search", *Proc. Eurospeech 93*, Berlin, September 1993.

[4] E. Giachin, P. Baggia, and G. Micca, "Language models for spontaneous speech recognition: a bootstrap method for learning phrase bigrams", *ICSLP 94*, Yokohama, Japan, September 1994.

[5] F. Jelinek, "Self-organized language modeling for speech recognition", 1987, in K.-F. Lee, A. Waibel (Eds.), *Readings in Speech Recognition*, Morgan-Kaufmann, 1989.

[6] H. Ney and U. Essen, "On smoothing techniques for bigram-based natural language modelling", *ICASSP 91*, Toronto, Ont., May 1991.

[7] B. Suhm, and A. Waibel, "Towards better language models for spontaneous speech", *ICSLP 94*, Yokohama, Japan, September 1994, pp. 831-834.