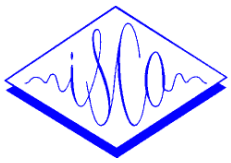

3rd International Conference on
Spoken Language Processing (ICSLP 94)
Yokohama, Japan
September 18-22, 1994

LANGUAGE MODELS FOR SPONTANEOUS SPEECH RECOGNITION: A BOOTSTRAP METHOD FOR LEARNING PHRASE BIGRAMS

Egidio Giachin *Paolo Baggia* *Giorgio Micca*

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

ABSTRACT

This study refers to the search for language models that are suitable for the recognition of spontaneous speech occurring in task-specific man-machine dialogue systems. Bigrams are an effective means for that purpose, however they only capture constraints between adjacent words. Task-specific training corpora are very expensive to collect and hence they are likely to be insufficient to reliably train trigrams. On the other hand, the type of sentences employed in these tasks are characterized by highly repetitive phrases that do occur enough times to suggest trying to automatically find and model them as if they were individual dictionary elements, so as to favor their recognition.

The determination of the word sequences to model is accomplished according to a perplexity minimization criterion, thus it is optimal insofar perplexity is a reliable quality measure for a language model. The procedure is iterative: starting with the original 1-word elements, it finds the pair of words for which the perplexity reduction is higher and connects them into a 2-word element. By cyclically continuing this action it "bootstraps" to longer-span elements, until no more perplexity reduction is obtained. Some variants of the algorithm are discussed and compared. This model produced a more than 20% perplexity reduction over 1-word bigrams, which makes it favorably comparable to trigrams.

1 INTRODUCTION

The use of statistical language models is a well known means for introducing additional constraints in a speech recognizer and hence improving its performance. The role of the language model is that of estimating the prior probability of the word sequences occurring in the task. One of the most successful approach is represented by the *bigram* model, which assumes that the probability of a word in a sequence only depends on the preceding word. A refinement of the bigram model, devised in order to cope with the sparseness of training data, is the *class bigram* model, where words are partitioned into equivalence classes (manually or automatically determined), and the inter-word transition probability is assumed to depend only on the word classes. This model gives rise to the well known formulas

$$P(w_i|w_{i-1}) = P(w_i|c_i)P(c_i|c_{i-1}) \quad (1)$$

and

$$P(w_1,\ldots,w_N) = \prod_{i=2}^{N} P(w_i|w_{i-1}) \quad (2)$$

for the word transition probability and the word sequence probability, where N is the sequence length. Bigrams (and more generally n-grams) were initially employed in tasks, such as dictation, that were not focused on specific semantic domains and for which large training corpora existed. Recent experiences show that they are also effective in limited-domain tasks such as those related to database enquiry services.

The limit of bigrams is their excessive locality, i.e. they only capture relationships between adjacent words; natural language is rich in structure and longer-memory models are necessary in order to account for it. This is especially true for the above mentioned tasks where a naive user is interacting with a spoken dialogue system to access information in a database, as in the EEC Sundial project [8]. Collecting a significantly large training corpus for such a task requires to have hundred of users interact with the system and is thus an extremely costly procedure. Typical corpora may have a number of words of the order of 100,000. The Italian corpus of spontaneous speech collected at Cselt, for example, includes 54634 words. Though further utterances may be added in the future, it is unlikely that corpora of such kind will ever be large enough to allow reliable training of trigrams. Hence different types of high-order language models have to be sought for.

In the above mentioned tasks the semantic domain is limited and the sentences people employ are full of repetitive phrases, either generic (such as *"I would like to know"*) or pertaining to the task (*"from Torino"*, *"at two fifteen"*, *"first class fare"*, *"what type of train is it"*, etc.), that intuitively seem to represent syntactic language constituents. These phrases may stand alone or be embedded in longer sentences, possibly corrupted by disfluencies or hesitations. In a previous study, it had been shown that phrases of this kind were associated to semantic concepts and could be automatically learned from a semantically annotated corpus, with application to understanding [5]. This suggests that having a simple model for repetitive phrases may help to reduce perplexity and hence to improve recognition. In the present study, frequent phrases are represented through arc sequences of a finite state grammar (see 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. In other words, phrases are treated as though they were single lexicon entries.

To determine which sequences should be represented this way, an entirely automatic procedure has been studied. The procedure works according to a perplexity minimization criterion, thus it is optimal insofar perplexity is a reliable quality measure for a language model. It starts by identifying the two words that, when connected into a single element, produce the highest perplexity reduction. By

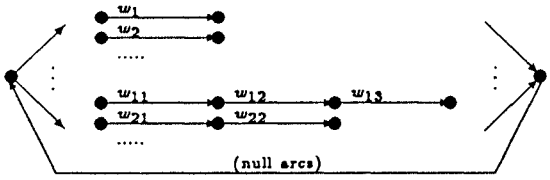

Figure 1: How word and phrase are represented

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.

Perplexity is taken as a measure of performance, which is evaluated on a set of 1202 sentences extracted from the global corpus. A perplexity reduction between 18% and 22% was obtained with respect to 1-word bigrams, depending on the lexicon and prior word clustering used, and a 5% reduction with respect to trigrams.

2 PERPLEXITY MINIMIZATION

The suggestion that frequent phrases composed of two or more words should be represented as a single lexicon entries and treated as such in bigrams is not a new one. A method for automatically identifying word chains was first proposed by R. Mercer in [6] and was based on occurrence countings. In the present study several information theoretic criteria for finding word associations have been tried, including some based on the computation of correlation among adjacent words. However, the criterion that seems most satisfactory is based on the minimization of an objective cost measure, namely perplexity [3] as measured on the training data.

The criterion acts in the following way:

1. Two words w_a and w_b are selected as candidate for connection into a phrase.
2. The training corpus is rewritten by replacing w_a and w_b with a new word w_{ab} whenever they occur in sequence.
3. Bigrams are estimated for the new set of words.
4. The difference is computed between the perplexity of the original corpus with the original word set, and the perplexity of the rewritten corpus with the new word set.

A positive difference means that combining the two words better accounts for the observed training corpus.

The perplexity Q of a word string $w_1, \ldots, w_N$ given a bigram model is determined as

$$Q = \exp(-\frac{1}{N}\hat{P}(w_1, \ldots, w_N)) \tag{3}$$

where

$$\hat{P}(w_1, \ldots, w_N) = \sum_{i=2}^{N} \ln P(w_i|w_{i-1}) \tag{4}$$

is derived from Eq. 2 and represents the estimated log-probability of the word sequence, given the language model. Hence minimizing Q amounts to maximizing the string probability, and probability is the actual quantity computed in point 4. A maximum likelihood approach of this kind has been used for several estimation problems in language modeling, including smoothing and automatic word clustering.

The determination of probability is computationally expensive but may be sped up if one considers that it is not necessary to take into account all the adjacent word pairs in the corpus to compute the new bigrams and the new probability, but only those including w_a, w_b, and their immediate neighbors. By suitably modifying occurrence counts, the necessary amount of computation may be considerably reduced. Note that, though the probability is computed on a "shrunk" corpus when some phrases have been replaced by single symbols, it is correct to always keep the original number of words N because this is the actual number of words that is seen by the recognizer. In the case where words had been partitioned into equivalence *classes* for the sake of statistical robustness, Eqs. 3 and 4 are still true, provided to replace words w_i with classes c_i and to add the class conditional term $P(w_i|c_i)$ according to Eq. 1. When a class phrase c_{ab} is created with classes c_a and c_b, the word usage is assumed to be independent of the fact that classes are connected together; hence the class conditional term is computed as

$$P(w_{ab}|c_{ab}) = P(w_a|c_a)P(w_b|c_b) \tag{5}$$

3 THE ALGORITHMS

The algorithms studied here consist of a base algorithm that just performs perplexity minimization according to the procedure outlined in the previous sections, and three variants. The basic algorithm is as follows:

BASE algorithm

1. Begin with a training corpus $\underline{w} = (w_1, \ldots, w_N)$, and the associated word class corpus $\underline{c} = (c_1, \ldots, c_N)$, resulting from having partitioned words among M possible classes out of a predefine set C.
2. Determine the two classes c_a, c_b that maximize

$$\underset{c_1 \in C, c_2 \in C}{argmax}\ \hat{P}(\underline{c}, c_1, c_2)$$

   where $\hat{P}(\underline{c}, c_1, c_2)$ is the log-probability of the modified training corpus when classes c_1 and c_2 have been connected;
3. Add the new class c_{ab}, deriving from the connection of c_a and c_b, to the set C, and modify the training corpus accordingly;
4. Loop to point 2 until $\hat{P}(\underline{c}, c_1, c_2)$ does not change from the previous iteration.

The classes representing unknown words and end-of-sentence symbols are never connected to any class. This algorithm is bound to converge. It belongs to the family of "greedy" algorithms and hence is not insured to converge to the global optimum. In particular, the following phenomenon occurs: In order to generate long word chains (e.g. *"what type of train is it"*), many shorter chains have to be generated first (e.g. *"what type of"*). Some of these shorter chains are no more useful after the longer ones have been generated, so they should be expunged and their original component words should be used instead. The following variant has then be implemented, which checks whether the deletion of any word chains may improve perplexity:

DEL variant

1. Begin with the output of the BASE algorithm (a corpus $\underline{c'} = (c_1, \ldots, c_{N'})$ and a set C' of M' classes).

2. Determine the two classes c_a, c_b that maximize

$$\underset{c_1 \in C', c_2 \in C'}{argmax} \; \hat{P}(\underline{c}', c_1, c_2)$$

where $\hat{P}(\underline{c}', c_1, c_2)$ is the log-probability of the modified training corpus when classes c_1 and c_2 have been *exploded* into their original component words;

3. Loop to point 2 until $\hat{P}(\underline{c}', c_1, c_2)$ does not change from the previous iteration.

Though even this algorithm cannot be insured to converge to the absolute optimum, experiments indicated that it does provide optimality in practice, and further refinements were not deemed necessary. (Incidentally, the number of deleted classes resulted very small.) Some other variants have nevertheless been implemented in order to improve the statistical robustness of the method. It has been noticed, for example, that it is useless to consider classes that occur a very low number of times. Hence the following variant checks that:

MIN variant

As the BASE algorithm; in point 2 c_1 and c_2 are not considered for grouping unless $counts(c_1, c_2) > T_{min}$.

Finally, a variant based on the leave-one-out method has been implemented. To compute the log-probability of the training corpus, a small portion is cyclically extracted from the corpus itself; the class counts are estimated on the retained portion, they are smoothed, and probability is computed on the extracted portion. We extracted one sentence at a time to avoid breaking of sentences that could make unrelated words to artificially appear contiguous. The so-called linear interpolation smoothing of [7] was employed. The variant is then

LOO variant

Similar to the BASE variant, but cyclically compute $\hat{P}(\underline{c}, c_1, c_2)$ on the held-out portion of the corpus, using counts estimated and smoothed on the retained portion.

4 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 [4]. Words were partitioned according to a set of 199 classes. The system was tested with "naive" users who received a short written explanation of the system's capabilities and then were let free to interact with the system itself in a completely unattended mode. A corpus of 8720 spontaneous speech utterances including 54634 words has been collected in this way [2]. This set was extended with an earlier set of 746 sentences, bringing the total number of words to 60089. The corpus includes long and complex sentences (many of them ungrammatical) as well as short and focused ones. A set of 1358 sentences has been extracted to perform testing. This set has been purged of sentences that contained speech disfluencies (e.g. interruptions) and out-of-dictionary words, because they could not have benefited much from the phrase bigram technique. The test set thus includes 1202 sentences.

Since the corpus was acquired through a dialogue system, sentences could be partitioned according to the dialogue state during which they were uttered. Four subcorpora were constructed in this way. These corpora were used to determine how the phrase bigram technique behaves when tried with groups of sentences that are semantically more focused.

Model		*Perplexity*
Bigram		34.2
Trigram		27.9
Phrase	$T_{min} = 20$	26.7
bigrams	$T_{min} = 50$	26.8
MIN version	$T_{min} = 80$	27.7

Table 1: Perplexity with the MIN version

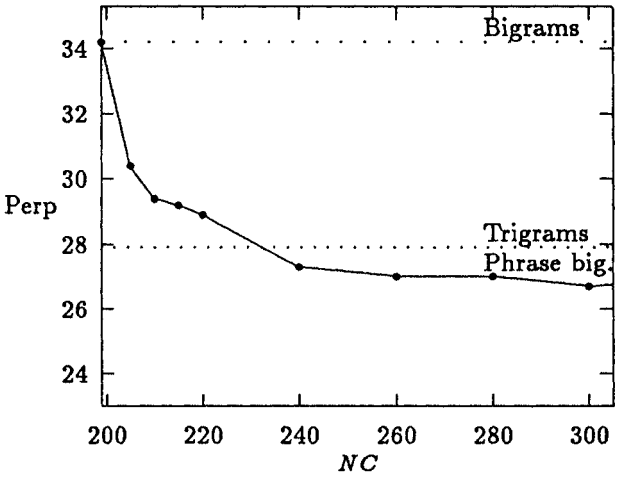

Figure 2: Perplexity reduction depending on NC

4.1 Global results

The different versions of the algorithms were compared on the basis of the perplexity they achieve on the test set. The BASE version was found to output a number of class phrases that was too high to be reliably trained. Thus in practice the MIN version has always been used. Table 1 compares the perplexity achieved by the MIN version with that of the bigram and trigram models obtained with the original 199 word classes. Three different values of minimum count threshold T_{min} are used. The table shows that a perplexity reduction of about 22% is achieved with respect to bigrams and of 4.5% to trigrams (which suffer from undertraining). It can also be seen that the perplexity obtained with different T_{min} values is about the same. This suggests to use high minimum count thresholds, so that statistical robustness is strengthened.

Fig. 2 refers to the MIN algorithm with $T_{min} = 20$. After the algorithm produced the new classes (appended to the original set of 199 classes), perplexity was computed for different subsets obtained by taking the first NC classes from this extended set. Results show that most of the perplexity fall is due to the first 40 or 50 classes found. Similar curves are obtained with other T_{min} values and with the other versions. These data show that the algorithm generates more classes than necessary, and that only a portion of them should be retained, which is an advantage for statistical robustness.

Table 2 shows a comparison between the other versions of the algorithm. A T_{min} of 50 was used in all cases. Data show that all versions provide similar performance. These results indicate that, at least for this kind of task, once the most relevant language constructs are captured, most of the work is done, and working in detail provides no further improvement.

Model	*Perplexity*
MIN	26.8
MIN+LOO	26.9
MIN+LOO+DEL	26.9

Table 2: Perplexity with different versions (T_{min} = 50)

State dep.	*Model*	*Place*	*Time*	*Info*	*Initial*
No	Bigram	37.8	41.9	29.0	45.0
	Trigram	25.4	39.5	23.6	40.7
	Phr.Big.	22.0	25.5	21.5	34.8
Yes	Bigram	17.8	27.3	26.6	40.7
	Trigram	17.8	27.3	24.2	36.9
	Phr.big.	17.1	25.6	21.8	35.0

Table 3: Perplexity with dialogue-state dependent training

4.2 Results in the dialogue context

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 the user focused questions. It is known that using state-dependent language models the recognizer performance is improved [1]. One might argue that the bootstrap phrase bigram method should also take advantage of that situation. This is confirmed by experiments. Training and test sentences were partitioned into four subcorpora according to what the user was asked to say:

Place A departure or arrival city or station.

Time A time expression (typically a departure time).

Info Any information on train services, fares, etc.

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

The above subcorpora are ranked from the most to the least focused. Table 3 compares the bigram, trigram, and phrase bigram models as tested for each of the four groups. Training has been done using the whole corpus (upper part of the table) or the state-dependent subcorpus (lower part). Data show that state-dependent training improved all models. For the more focused case (*Place*) the phrase bigrams perform better than when they are trained trained on the whole corpus (they also perform better than any of the other models). However for the more generic cases (*Time*, *Info*, and *Initial*) there is no such improvement. By taking the best models for each subcorpus an overall perplexity of 24.6 is obtained, a 15% and an 8% improvement over state-dependent bigrams and trigrams, respectively.

4.3 Recognition results

The integration of phrase bigrams into a Viterbi-like recognizer is that of a stochastic finite-state grammar [4]. Current experimentation employs 310 context dependent subword units, modeled through 3-state discrete density HMMs. The set of acoustic units was defined through occurrence counts performed on an initial training set of read sentences; training on the spontaneous speech corpus, and extension to continuous density HMMs, are currently being carried out. First recognition results indicate a 12% decrease of word error rate on the test set.

5 CONCLUSIONS

A method to overcome the limit of bigrams has been investigated, which is less sensitive than trigrams to sparseness of training data. The method is based on the use of bigrams of phrases as well as words. Phrases to be modeled are automatically determined by a procedure that follows a perplexity minimization criterion. When tested on spontaneous speech sentences referring to a train timetable enquiry dialogue task, this approach achieved a more than 20% perplexity reduction over standard bigrams.

A look at the phrases found by the procedure is interesting. Some are mere groups of words that frequently occur in sequence. Most of them, however, are sensible phrases pertaining to the task and representing linguistic constituents. Examples include phrases on locations (*"from Torino to Milano"*), train services (*"are there sleeping cars"*), courtesy requests (*"I would like to know"*), etc. Though finding language constituents was beyond the goal of this work and does not necessarily improve recognition accuracy per se, it might be useful to help designing language models for parsing and understanding.

References

[1] F. Andry, "Static and dynamic predictions: a method to improve speech understanding in cooperative dialogues", *ICSLP 92*, Banff, Alberta, 1992.

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

[3] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition", *IEEE Trans. on PAMI*, Vol. 5, March 1983.

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

[5] E. Giachin, "Automatic training of stochastic finite-state language models for speech understanding", *ICASSP 92*, San Francisco, March 1992.

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

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

[8] J. Peckham, "A new generation of spoken language systems: recent results and lessons from the Sundial project", *Eurospeech 93*, Berlin, September 1993.