

DP-BASED WORDGRAPH PRUNING

Thomas Kuhn, Pablo Fetter, Alfred Kaltenmeier, Peter Regel-Brietzmann

Daimler-Benz AG, Research and Technology,
Speech Understanding Systems (F3M/S)
Wilhelm-Runge-Str. 11, 89081 Ulm, Germany
e-mail: kuhn@dbag.ulm.DaimlerBenz.com

1. ABSTRACT

In this paper we present an efficient technique of generating word graphs in a continuous speech recognition system. The word graph is constructed in two stages. In the first stage, a huge word graph is generated as a by-product of a beam-driven forward search. Based on a Dynamic-Programming (DP) method, this huge word graph will be pruned in the second stage using higher level knowledge, such as n -gram language models. In this pruning stage an edge is removed if there is no path going through this edge which is better scored as the best-scored path in the word graph. The proposed technique will be evaluated in the German VERBMOBIL task.

2. INTRODUCTION

Speaker-independent large-vocabulary speech recognizer in spoken dialog systems require a powerful linguistic postprocessor. The interface between the speech recognizer and the linguistic analysis is usually a word graph [Cla93a, Oer93, Wah93] or a list of the n best scored sentence hypotheses [War95, Gla95]. A word graph consists of a set of edges where each edge represents a scored word hypothesis. The more edges in the word graph, the higher the coverage; but on the other side the computational effort in the linguistic analysis is also higher since more possible paths have to be investigated by the linguistic component.

Compared to an n -best interface, the word graph structure is more flexible, since even word graphs with a small density can achieve a high coverage [Oer93]. The n -best approach suffers from two main drawbacks. First, the number n of generated sentences depends directly on the length of the utterance. The longer the

utterances, the more sentences to be generated. Second, most times sentences in an n -best list differ only in one or two words, while the other words remain unchanged. Due to this fact and especially for long utterances, a large number of sentences must be generated to obtain the same coverage as a moderate sized word graph.

In this paper, we will present a new technique of word graph generation which works in two or more subsequent stages. A description is given in the following sections. First, the proposed DP-based Graph-Pruning (DP-GP) technique is described in detail. Then, a short sketch of our acoustic-phonetic modeling is given. Finally, we evaluate the DP-GP method in the German VERBMOBIL task [Wah93].

3. THE DP GRAPH-PRUNING ALGORITHM

In this section we will give a formal description of the DP Graph-Pruning technique (DP-GP). A word graph \mathcal{G} is a quintuple $g_j = (a(j), e(j), t_a(j), t_e(j), \gamma(j))$, where $a(j)$ and $e(j)$ are the start and end nodes, $t_a(j)$ and $t_e(j)$ terms the corresponding time frames of the start and end nodes, and $\gamma(j)$ the log HMM score of the word w_j in the region of $[t_a(j), t_e(j)]$. Figure 1 shows

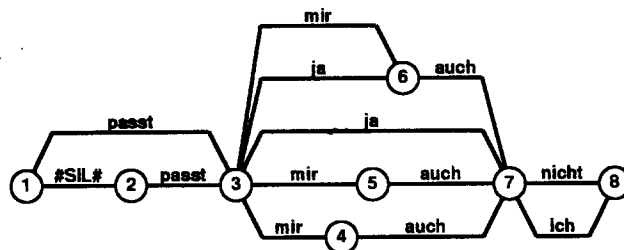


Figure 1: Example of a word graph

This work was partly supported by the German Federal Ministry of Education, Science, Research and Technology.

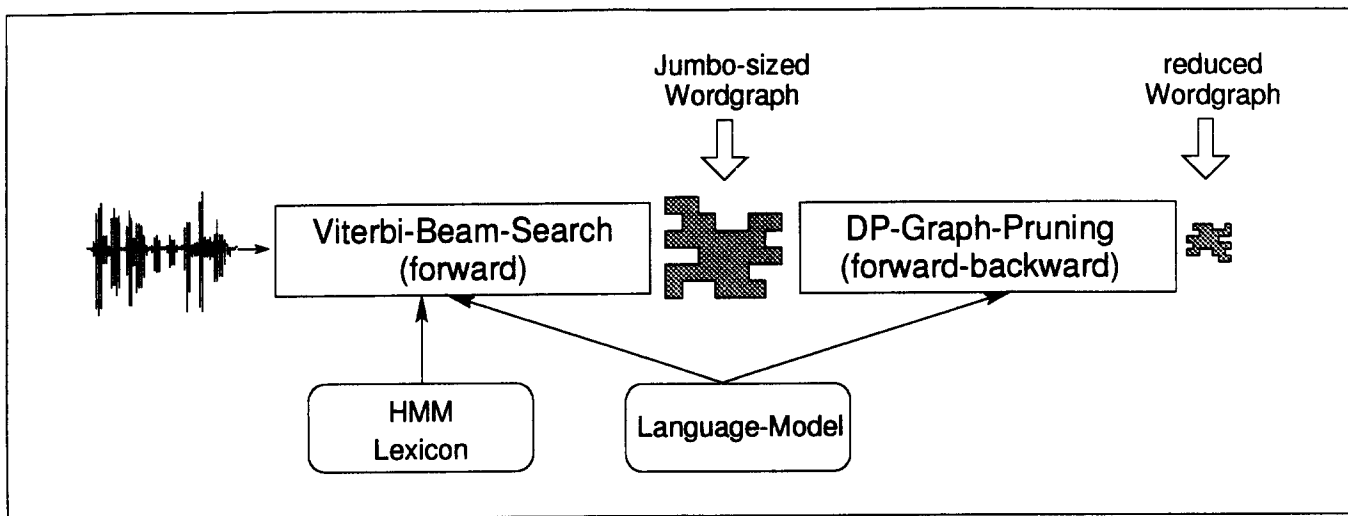


Figure 2: Two stage architecture for word graph generation in a word recognizer

an example of a word graph, which was generated to the spoken utterance "das passt mir auch nicht".

Similar to the algorithm described in [Mur93], the word graph generation is constructed in two stages (see Figure 2). The main idea is to generate a huge word graph in the first stage which will be pruned in the second stage using higher level knowledge, such as n -gram language models. For simplicity, the following description is given for a bigram model.

In the first stage, the word graph is generated as a by-product of a beam-driven forward search. For efficiency reasons, the lexicon is realized as a tree. In contrast to the time-synchronous word graph generation technique described in [Oer93], we use an asynchronous method for word graph generation. If the last state of a word is reached, a new lexicon tree is initialized in a certain region. This lexicon tree is initialized by the best scored path ending in this time frame. The word graphs produced should be large enough, so that all uttered words are included as word hypotheses. It should be noted, that word hypotheses which were removed in the first stage, cannot be recovered in the second stage. Furthermore, in the first stage, the information of a language model is only used to determine the m best scored word hypotheses. m is a predefined constant, adjusting the size of the word graph in the first stage.

In the second stage, the so called "jumbo sized word graph" is reduced significantly, so that the following linguistic analysis is able to parse the (smaller) word graph within a reasonable computation time. The second stage is based on a dynamic programming procedure. For each edge g_j in the word graph \mathcal{G} the score of

the best path which passes through this edge is computed by a forward and a backward path. The forward path is computed recursively from left to right as follows:

$$\alpha_{e(j)}(j) = \min_{g_i: a(j)=e(i)} (\alpha_{e(i)}(i) + \gamma(j) - \vartheta \cdot \log(P(w_j | w_i)))$$

where ϑ is a linguistic matching factor to weight the influence of the language model. $\alpha_{e(j)}(j)$ denotes the best scored forward path passing edge g_j in end node $e(j)$. A path through the word graph can only be extended, if the end node $e(i)$ of the last edge g_i in the path is identical to the start node $a(j)$ of the next edge g_j .

A similar recursion has to be computed for the backward path $\beta_{a(j)}(j)$, but as opposed to the α values the β values are indexed by the start node $a(j)$ of the word hypothesis g_j . The recursion is done from right to left as follows:

$$\beta_{a(j)}(j) = \min_{g_i: e(j)=a(i)} (\beta_{a(i)}(i) + \gamma(j) - \vartheta \cdot \log(P(w_i | w_j)))$$

In the pruning step all edges $g_i, g_j \in \mathcal{G}$ that fulfill the following restriction are discarded:

$$\alpha_{e(j)}(j) + \beta_{a(i)}(i) - \vartheta \cdot \log(P(w_i | w_j)) > \frac{1}{\theta} \cdot \alpha^*$$

with $e(j) = a(i)$

where α^* denotes the best scored path in the word graph \mathcal{G} determined by the forward path. θ terms a threshold in the range between zero and one. A

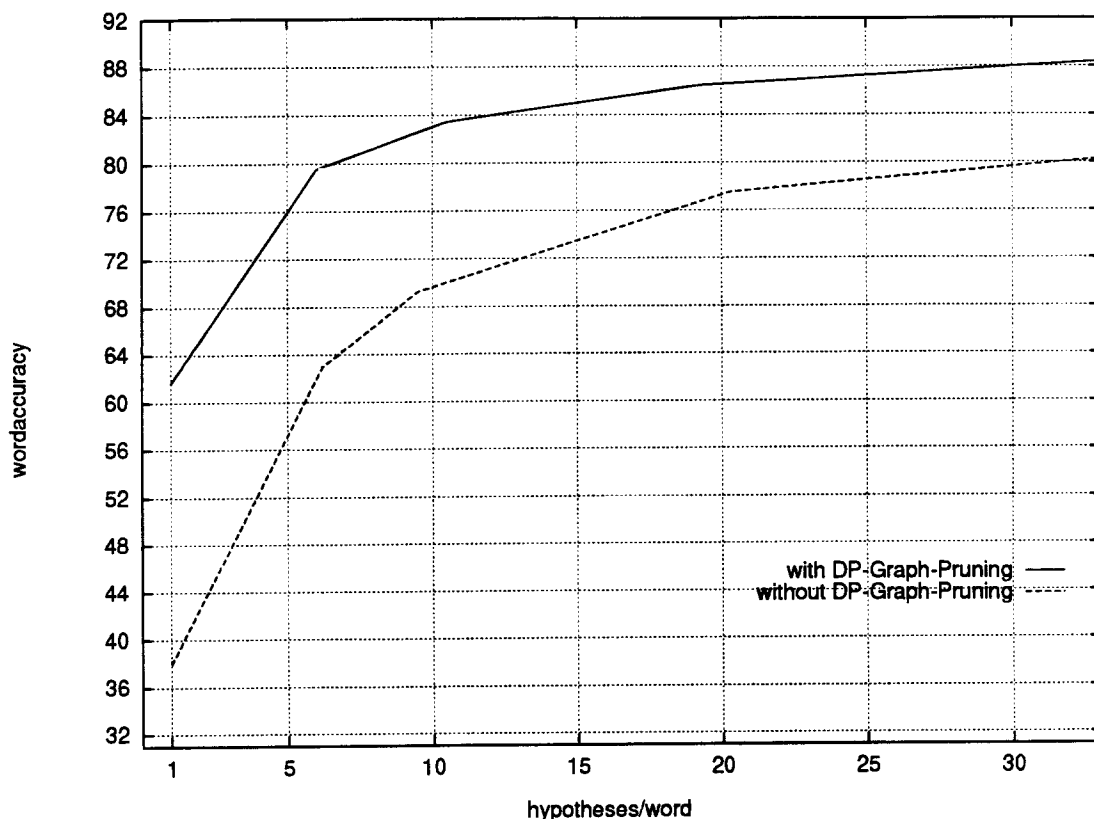


Figure 3: Results for the VERBMOBIL database using the DP-GP method.

value close at one means that nearly all edges in the word graph are discarded, whereas a value close to zero means that only a few word hypotheses are removed.

4. ACOUSTIC-PHONETIC MODELING

The speech signal is sampled at 16 kHz, quantized with 16 bit and partitioned into 10 msec frames. Applying a 512 FFT, a vector of 13 cepstral features is computed for each frame. The cepstral features are normalized to remove spectral influences of the speaker, microphone, room acoustics, and transmission line. After that, a LDA is applied to each frame with a four frame neighbourhood resulting in a 32-dimensional feature vector. Considering a four frame neighbourhood for LDA computation, we don't use first and second derivatives of the cepstral features. Subword units include 1030 triphones and 115 whole-words modeled by semi-continuous HMMs. The training of HMM parameters is done using a multi-stage algorithm which is described in [Cla93b] in detail. On a Dec AlphaStation 200 with 233 MHz, the real time factor for the word recognizer considering a vocabulary with 3 305 is less than 3.

5. RESULTS AND DISCUSSION

We performed experiments on the German VERBMOBIL [Wah93] database (also called the German Spontaneous Scheduling Task), which consists of about 200 human-to-human spontaneous speech dialogs. In all of these dialogs, two persons were trying to arrange a date for a meeting. The dialogs have been collected and transcribed at various German universities. The whole corpus contains more than 100 000 words in 5 000 utterances, including phenomena such as pronunciation variations, word fragments, etc. The test set consists of 331 utterances or about 40 minutes of speech. The vocabulary of the word recognizer consists of 3 305 different words. The perplexity of bigram model is about 68.

In the first experiment we evaluated only the word graphs produced in the first stage. In the second experiment we applied the DP-GP method to these word graphs, starting with a word graph density of about 100 hypotheses per word. The word graph density was adjusted by the threshold θ . The results are shown in Figure 3. It can be seen that the DP-GP method improves the recognition performance significantly. For

the best word chain, we achieved about 62% word accuracy. If 6 word hypotheses per word are generated, the word accuracy increases to about 80%. If 10 word hypotheses per word are generated – this is currently the largest density which can be handled by the linguistic analysis – the word accuracy increases further to 83%.

6. CONCLUSIONS

We presented an efficient two-stage algorithm for word graph generation. In the first stage a huge word graph is generated using a beam-driven viterbi search. This word graph is drastically reduced in the second stage using the proposed DP-based pruning method. It was shown, that the performance of word recognizer could be increased significantly. Using a standard bigram in the second stage and considering a density of 6 hypotheses per word, the word accuracy increased from 63% to 80%. It should be noted, that also longer spanning n -grams such as trigrams or polygrams (see [Kuh94]) can easily be integrated.

7. REFERENCES

- [Cla93a] F. Class, A. Kaltenmeier, and P. Regel-Brietzmann. Evaluation of an HMM Speech Recognizer with various Continuous Speech Databases. In *Proc. European Conf. on Speech Technology*, pages 807–810, Berlin, 1993.
- [Cla93b] F. Class, A. Kaltenmeier, and P. Regel-Brietzmann. Optimization of an HMM-Based Continuous Speech Recognizer. In *Proc. European Conf. on Speech Technology*, pages 803–806, Berlin, 1993.
- [Gla95] J. Glass, D. Goddeau, L. Hetherington, M. McCandless, C. Pao, M. Phillips, J. Polifoni, S. Seneff, and V. Zue. The MIT ATIS System: December 1994 Progress Report. In *Spoken Language Systems and Technology Workshop*, pages 252–256, Morgan Kaufmann, 1995.
- [Kuh94] T. Kuhn, H. Niemann, and E. G. Schukat-Talamazzini. Ergodic Hidden Markov Models and Polygrams for Language Modeling. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 357–360, Adelaide, Australia, 1994.
- [Mur93] H. Murveit, J. Butzberger, Digalakis V., and Weitraub M. Large-Vocabulary Dictation using SRI's DECIPHER Speech Recognition System: Progressive Search Techniques. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 319–322, Minneapolis, 1993.
- [Oer93] M. Oerder and H. Ney. Word Graphs: An Efficient Interface between Continuous-Speech Recognition and Language Understanding. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 119–123, Minneapolis, 1993.
- [Wah93] W. Wahlster. Verbmobil — Translation of Face-to-Face Dialogs. In *Proc. European Conf. on Speech Technology*, volume “Opening and Plenary Sessions”, pages 29–38, Berlin, 1993.
- [War95] W. Ward and S. Issar. The CMU ATIS System. In *Spoken Language Systems and Technology Workshop*, pages 249–251, Morgan Kaufmann, 1995.