GENERATION AND EXPANSION OF WORD GRAPHS USING LONG SPAN CONTEXT INFORMATION

Christoph Neukirchen, Dietrich Klakow, Xavier Aubert

Philips Research Laboratories, Weisshausstr. 2, 52066 Aachen, Germany
Email: {Christoph.Neukirchen, Dietrich.Klakow, Xavier.Aubert}@philips.com

ABSTRACT

A new algorithm for the generation of word graphs in a cross-word decoder that uses long span m-gram language models is presented. The generation of word hypotheses within the graph relies on the word m-tuple-based boundary optimization. The graphs contain the full word history knowledge information since the graph structure reflects all LM constraints used during the search. This results in better word boundaries and in enhanced capabilities to prune the graphs. Futhermore the memory costs for expanding the m-gram constrained word graphs to apply very long span LMs (e.g. ten-grams that are constructed by log linear LM combination) are considerably reduced. Experiments for lattice generation and rescoring have been carried out on the 5K-word WSJ task and the 64K-word NAB task.

1. INTRODUCTION

Word graphs are used in speech recognition systems to represent the most probable recognition hypotheses in a compact way. Hypotheses in a graph, which is commonly generated by using computationally cheaper models, can be rescored by the application of more expensive knowledge sources, e.g. long span language models or natural language understanding models [6, 7].

In the bigram language model (LM) decoding framework [7, 1] the word-pair approximation [9] for the dependence of the word boundaries plays an important role for constructing word graphs. The boundary assumption implicitly results from the dynamic programming (DP) search strategy that is conditioned on the predecessor word. Such boundary optimization generates each word-pair at most once at any ending time and thus keeps the graph relatively size small by avoiding redundant sentence hypotheses.

An alternative way to construct word graphs is based on the DP recombination of time-conditioned hypotheses [6, 8]. By construction, this method does not rely on the word-pair approximation. In general, the time-conditioned method produces very large word graphs with many redundant hypotheses. Hence, these graphs have to be reduced in a subsequent step by an explicit boundary optimization on the word level.

In the newest time-synchronous beam search decoders [5, 2] long span context dependent models like cross-word HMMs and m-gram ($m > 2$) LMs are integrated. As shown in the following sections, using these enhanced knowledge sources leads to the generation of improved word graphs that are based on an extended word m-tuple boundary assumption. In addition, the structural graph constraints allow an efficient expansion of word graphs to further apply very long span history dependent knowledge sources like ten-gram LMs.

2. DECODING

The generation of word graphs is integrated in a time-synchronous one-pass DP-decoder [2] that uses cross-word HMMs and m-gram LMs. The DP works on partial hypotheses that are conditioned on their $(m-1)$-word predecessor history according to the LM m-gram order. The word lexicon is organized as a reentrant phonetic prefix tree. The following definitions are introduced to describe the word graph generation:

$\mathbf{W}$: an N word sequence $\mathbf{W} = (w_1, \ldots, w_N)$.

$S_m(\mathbf{W})$: the m-gram LM-state of a sequence $\mathbf{W}$ is given by the $(m-1)$ most recent words $S_m(\mathbf{W}) = (w_{N-m+2}, \ldots, w_N)$.

$h(w;\tau,t)$: probability $p(\mathbf{X}_{\tau+1}^t|w)$ that word w produces the acoustic vectors $\mathbf{X}_{\tau+1}^t = x(\tau+1), \ldots, x(t)$.

$G(\mathbf{W};t)$: joint probability $p(\mathbf{X}_1^t|\mathbf{W}) \cdot P(\mathbf{W})$ of generating the acoustics $\mathbf{X}_1^t$ *and* a word sequence $\mathbf{W}$ with ending time t.

$H(S_m;t)$: joint probability of generating the acoustics $\mathbf{X}_1^t$ *and* a word sequence with the final $(m-1)$ words given by S_m at ending time t.

As shown in [4] the definition of the m-gram state $S_m(\mathbf{W})$ must be augmented by the phonetic fan-out context of $\mathbf{W}$ for decoding and generation of word graphs with cross-word acoustic models.

2.1. Recombination of partial search hypotheses

The DP optimization in the decoder is conditioned on the partial hypotheses' $(m-1)$ word history S_m. When the search reaches the leaf for word w in the lexicon tree this results in an extension of the preceding partial hypothesis $\mathbf{W}$ to the new hypothesis $\tilde{\mathbf{W}} = (\mathbf{W}, w)$. Within the lexical tree, the DP recombination is applied to all preceding hypotheses being in an identical m-gram state $S_m(\mathbf{W})$ but having entered the tree at different starting times τ. Using the m-gram $P(w|S_m(\mathbf{W}))$, the following optimization generates the probability for the new partial hypothesis $\tilde{\mathbf{W}}$ ending at time t:

$$G(\tilde{\mathbf{W}};t) = P(w|S_m(\mathbf{W})) \cdot \max_{\tau}\{H(S_m(\mathbf{W});\tau) \cdot h(w;\tau,t)\} \tag{1}$$

The optimal tree starting time (boundary between $\mathbf{W}$ and w) τ_{opt} is given implicitly by the optimization in Eq. 1, and the boundary only depends on the ending time t and the m most recent words in $\tilde{\mathbf{W}}$, hence $\tau_{opt} = \tau(S_{m+1}(\tilde{\mathbf{W}});t)$.

3. WORD GRAPH ALGORITHM

The word graph to be generated consists of edges corresponding to word hypotheses w containing this word's acoustic and LM scores. The graph edges connect nodes which are associated with the word boundary times and the local hypotheses' m-gram states: all partial paths ending in the same graph node have an identical $(m-1)$ word history (for cross-word models the history is extended by the phonetic fan-out context [4]).

3.1. Word graph generation

The following algorithm constructs a m-gram constrained word graph using the hypotheses information provided by the decoder:

1. At each time t, consider all word m-tuples within the beam:

   $$\underbrace{(u,\ldots,v,w)}_{m\ words}$$

   By Eq. 1, at time t each word m-tuple is generated by the decoder at most once.

2. For each $(u,\ldots,v,w;t)$ keep track of the word boundary $\tau(u,\ldots,v,w;t)$ provided by Eq. 1.

3. Create a graph edge that contains:

   - the current word w
   - the word acoustic score $h(w;\tau(u,\ldots,v,w;t),t)$
   - the word m-gram LM-score $P(w|u,\ldots,v)$

4. For each individual pair of time and m-gram LM-state $(t;S_m(u,\ldots,v,w))$ create a graph node.

5. Link the graph edge of $(u,\ldots,v,w;t)$ with

   - the start node $(\tau(u,\ldots,v,w;t);S_m(u,\ldots,v))$
   - the end node $(t;S_m(u,\ldots,v,w))$

6. Path management:
   The beam pruning strategy and hypothesis recombination prevent hypotheses from being further expanded in the DP. Unexpanded hypotheses can cause dead paths in the word graph that do not reach the final graph node. A garbage collection removes dead partial paths to reduce memory consumption, at regular intervals.

3.2. Word graph properties

In step 5 of the word graph algorithm all partial hypothesis paths at the same ending time being in the same m-gram-state are merged into a single node before the DP recombination is applied in the decoder. When the best path through this node is extended in one of the subsequent steps of the algorithm, the dependence of the boundary time associated with this node is confined to the identity of the m most recent words as given by step 2. Hence, for all partial sentences ending at time t in the graph and sharing the same m recent words an identical word boundary τ is assumed.

Such word m-tuple assumption for the word boundary can be regarded as an extension of the word-pair approximation in [9]. Although the use of the word-pair approximation may be questionable for short words [8] it works satisfactory in most cases [1]. With the increasing m-gram order used in the word graph construction algorithm the boundary is further improved due to the extended context length taken into account in Eq. 1.

In Fig. 1 examples of search hypotheses and the corresponding word graphs in a bigram and a trigram decoder are shown. Using longer span LMs (here: trigram vs. bigram) delays the recombination and more LM-states will occur, in general. In the corresponding word graph this leads to the generation of more nodes and to lower branching factors (assuming an equal number of word hypotheses). When a partial hypothesis is not further expanded due to recombination or pruning (see Fig. 1) the resulting dead path will be typically longer for the trigram graph because of the lower branching factor. So, the final trigram word graph contains fewer edges and has smaller branching factors compared to the bigram graph.

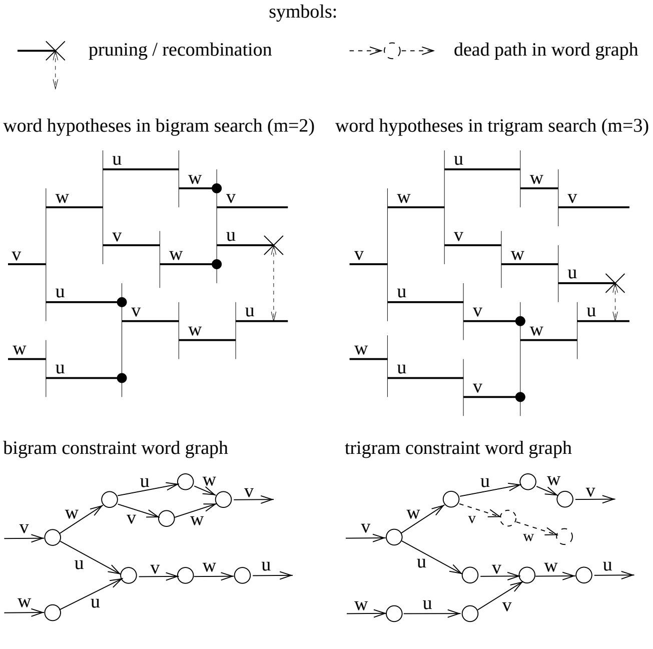
symbols:
pruning / recombination
dead path in word graph
word hypotheses in bigram search (m=2)
word hypotheses in trigram search (m=3)
u
w
v
bigram constraint word graph
trigram constraint word graph

Figure 1: Generation of m-gram constraint word graphs using bigram ($m = 2$) and a trigram ($m = 3$) decoding.

4. LONG SPAN HISTORY WORD GRAPH EXPANSION

To rescore the hypotheses within the graph using longer span history conditioned models (e.g. n-gram LMs, $n > m$) the m-gram graph structure has to be expanded according to the new context length. Typically the resulting graph size is increased exponentialy with $(n - m)$. Thus, the incorpation of long span conditioned models (i.e. larger m) in the word graph construction can reduce the costs for expanding the graph considerably. Furthermore, when the goal is to find the top-best hypothesis given the new long span LM the graph expansion can be performed localy at each DP optimization step [1].

4.1. Construction of long range language models

Backing-off is unsuitable for very long-range LMs. Techniques combining language models of different range and type perform much better. Log-linear interpolation [3] can be viewed as a simplified version of maximum-entropy models and is defined by

$$p(w|h) = \frac{1}{Z_\lambda(h)} \prod_i p_i(w|h)^{\lambda_i} \qquad (2)$$

where p_i are the different components to be combined and $Z_\lambda(h)$ is the normalization, which is however ignored in recognition.

In this paper log-linear interpolation is used for 6-gram and 10-gram LMs. For the 10-gram we have the specialized structure

$$p(w|h) = \frac{1}{Z_\lambda(h)} \quad p_5(w|h)^{\lambda_5} \prod_{i=0}^{8} \left(\frac{p_2^i(w|h_{-i})}{p(w)} \right)^{\lambda_2^i} \prod_{j=3}^{7} \left(\frac{p_3^j(w|h_0 h_{-j-1})}{p(w|h_0)} \right)^{\lambda_3^j} \tag{3}$$

where h_0 is the word immediately preceding w and all older words of the history have negative indexes. The 6-gram has the same structure but the last four terms of the products are skipped.

The components are:

- $p_5(w|h)$: A standard backing-off five-gram smoothed additionally by linear interpolation with a class-five-gram. The count-trees of both models are pruned and the relative size is adjusted to optimize perplexity. The exponent is $\lambda_5 = 0.85$.
- $p_2^i(w|h_{-i})$: Distance-i-bigrams where intermediate words in the history are removed, e.g. a distance-1-bigram ignores h_0 and instead goes one word further in the history. A distance-8-gram adds 10-gram information to the combined model. For distances up to four individual count-trees are used. The distance-5-bigrams to distance-8-bigrams share a count-tree. All distance-bigrams are a linear interpolation of word and class models. The weight of the distance-8-bigram is $\lambda_2^8 = 0.23$.
- $p(w)$: A unigram is needed to cancel out the unigram information always present in a distance-bigram.
- $p_3^j(w|h_0 h_{-j-1})$: Distance-j-trigrams. It is a bigram extended by an additional older word h_{-j-1} from the history. All distance-trigram access the same tied count-tree.. The weight of the distance-8-bigram is $\lambda_2^8 = 0.19$.
- $p(w|h_0)$: A bigram that cancels out irrelevant parts of the distance-trigrams.

Tab. 1 gives the perplexities. Note the decreasing perplexity gains from 3-gram to 6-gram and from 6-gram to 10-gram, but the total improvement of the 10-gram compared to the trigram is still 28%.

LM-Range	2	3	4	6	10
PP	112.7	60.4	50.4	45.4	43.3

Table 1: Perplexities for an increasing language model range.

5. EXPERIMENTAL RESULTS

The recognition system uses gender-dependent triphone HMMs trained on the WSJ0+1 database. Results are presented for a 5K and for a 64K vocabulary task. The 5K task consists of the male speakers of the Nov'92 and Nov'93 evaluation and development sets (776 sentences, 13113 spoken words, OOV: 0.16%). The 64K task consists of the female speakers in the Nov'94 development and evaluation sets of the North American Business (NAB) corpus (315 sentences, 7770 spoken words, OOV: 0.73%). In all experiments wide beam widths are used in the search to generate large word graphs.

The acoustic search space is specified by the following quantities: the average number of *word end* hypotheses and of *LM-states* per time frame. Word end hypotheses are the potential edges in the graph; the potential nodes in the graph are derived from the LM-states. *Word error rates* (WER) are given for first-pass decoding and for LM-rescoring experiments.

To specify the properties of the generated graphs the following quantities are given [4]: average *word graph density* (WGD), *word graph branching factor* (BF), and *graph word error rate* (GER) of matching the spoken sentence.

5.1. Word graph generation

The generation of word graphs using various kinds of acoustic models and LMs is illustrated in the left hand parts of Tab. 2 for the 5K task and of Tab. 3 for the 64K task. The number of word end hypotheses and of different LM-states in the search space increases when moving to longer span LMs and to cross-word HMMs caused by the extended context dependencies. The top-best WER of the search is improved due to the better knowledge sources. For longer m-gram contexts a decrease of the BF in the word graphs can be observed that directly corresponds to the decreasing ratio between the number of word ends and LM-states. Although the decoder generates more word-ends for longer span context models, the sizes (WGD) of the resulting m-gram word graphs are significantly smaller. As explained in Sec. 3.2, this apparently paradoxical situation can be attributed to the increasing average length of dead paths in addition to the stronger pruning using more detailed knowledge sources. Hence, the size of the m-gram graphs is reduced by the hypotheses with small probabilities with increasing m-gram order. With the lower graph density and the smaller branching factor, the total number of sentence hypotheses contained in the graph is reduced which increases the graph word error rate (GER) in spite of improved top-best WER.

5.2. Word graph pruning

To control the final word graph size, forward-backard (FB) pruning [5] with different beam width settings (see first column Tab. 2 and 3) is applied. This method efficiently makes use of all acoustic and LM knowledge sources contained in the m-gram word graphs and pruning is based on Thus, the most probable hypotheses are always kept within the pruned graphs. FB-pruning clearly takes advantage from the improved long span context knowledge sources in the higher order m-gram graphs: after pruning down the word graphs to comparable sizes, the GER is improved for the longer span LM and cross-word graphs. This effect becomes more significant for smaller target sizes of the graphs.

5.3. Rescoring with longer span language models

On the right hand parts of Tab. 2 and 3 rescoring results are shown for applying $n = 4, 6, 10$-gram LMs to the WSJ m-gram word graphs and $n =$ 3-gram LM to the NAB m-gram graphs. Significant performance improvements can still be observed when moving from long span history dependent models used in the decoder to very long span LMs for rescoring. The rescoring WERs on the original unpruned graphs (top line in each box) are not sensitive to the m-gram order of the graphs. This indicates that the relevant top-best rescoring sentence hypotheses are contained in almost all m-gram graphs. Furthermore the influence of the word boundaries in the m-gram graphs that are optimized according to the m-tuple approximation on the performance is small. Only for the WSJ within-word bigram constraint word graph in some cases small degradations may be attributed to non-optimal boundaries; any context-conditioned boundary optimization beyond the word-pair approximation results in the optimal rescoring WER. The rescoring WERs on the FB-pruned word graphs demonstrate

FB	word graph properties			n-gram rescoring WER		
beam	WGD	BF	GER	n=4	n=6	n=10
2-gram word graph (m=2), within-word HMM search: word ends: 291, LM-states: 61, WER: 8.14%						
-	4635.7	9.10	0.29%	5.27%	4.96%	4.82%
150k	146.7	3.03	0.43%	5.27%	4.96%	4.82%
100k	36.5	2.39	0.78%	5.27%	4.99%	4.89%
50k	10.2	1.85	2.04%	5.34%	5.12%	5.00%
20k	4.0	1.47	4.74%	6.22%	6.13%	6.05%
3-gram word graph (m=3), within-word HMM search: word ends: 397, LM-states: 206, WER: 5.85%						
-	1900.2	3.81	0.40%	5.23%	4.96%	4.77%
150k	138.6	2.33	0.50%	5.23%	4.96%	4.79%
100k	39.2	2.05	0.79%	5.25%	4.96%	4.79%
50k	10.7	1.75	1.98%	5.22%	4.91%	4.79%
20k	4.1	1.45	3.65%	5.33%	4.85%	4.94%
4-gram word graph (m=4), within-word HMM search: word ends: 294, LM-states: 214, WER: 5.23%						
-	301.5	2.84	0.75%	5.23%	4.96%	4.77%
150k	117.5	2.24	0.78%	5.23%	4.96%	4.77%
100k	50.5	1.96	0.91%	5.23%	4.96%	4.77%
50k	12.5	1.71	1.77%	5.23%	4.91%	4.71%
20k	4.4	1.43	3.25%	5.23%	4.85%	4.70%
2-gram word graph (m=2), cross-word HMM search: word ends: 1354, LM-states: 625, WER: 6.85%						
-	707.5	4.90	0.72%	4.64%	4.42%	4.34%
150k	79.4	2.54	0.75%	4.64%	4.43%	4.34%
100k	28.9	2.16	1.06%	4.64%	4.46%	4.34%
50k	9.2	1.77	2.08%	4.73%	4.55%	4.46%
20k	3.6	1.41	4.00%	5.27%	5.22%	5.20%
3-gram word graph (m=3), cross-word HMM search: word ends: 1459, LM-states: 952, WER: 5.08%						
-	231.2	3.00	1.08%	4.84%	4.48%	4.36%
150k	59.6	2.23	1.09%	4.84%	4.48%	4.36%
100k	26.9	1.99	1.27%	4.84%	4.48%	4.36%
50k	9.4	1.71	2.08%	4.84%	4.52%	4.42%
20k	3.6	1.40	3.46%	4.79%	4.61%	4.53%

Table 2: 5K WSJ task: properties of m-gram constrained word graphs and rescoring results for n-gram LMs.

that the graph sizes can be considerably reduced (down to a WGD of 30) without any degradations on the n-gram rescoring performance. When the word graph sizes are further reduced, there is a clear advantage for rescoring on longer range context dependent graphs: due to the improved pruning properties the degradation of rescoring WER becomes smaller when increasing the word graph context constraint order. This indicates that the best n-gram rescoring hypotheses obtained from the heavily pruned higher order word graphs are not contained in the lower order graphs of comparable size.

6. CONCLUSION

An extended method for generating word graphs using long span LMs and cross-word HMMs has been described. The advantages for using higher order context conditioned word graphs are: i) better modeling of word boundaries due to an extended word m-tuple boundary optimization; ii) improved pruning of word graphs by exploiting the more detailed knowledge sources; iii) smaller costs for graph expansion since higher order context constraints are encoded in the word graph structure. Evaluations on LVCSR tasks show that rescoring of m-gram graph using very long span LMs leads to useful WER improvements. An apparent disadvantage of using long range context dependent word graphs is that the number of hypotheses contained in the graphs is reduced. This has, however extremely small influence on the rescoring performance. The minor influence may be also attributed to relatively *similar* LMs (2-gram to 10-gram trained on the same data) used here, that exploit *similar* top best hypotheses in the graphs; the situation may change when applying knowlege sources that are different in nature, e.g. related to speech understanding.

FB	word graph properties			rescoring WER
beam	WGD	BF	GER	trigram (n=3)
2-gram word graph (m=2), within-word HMM word ends: 311, LM-states: 100, WER: 12.54%				
-	1896.8	9.20	1.62%	9.47%
150k	193.4	3.61	1.70%	9.47%
100k	53.4	2.64	2.19%	9.49%
50k	12.1	1.94	4.12%	9.63%
20k	4.3	1.56	7.66%	10.32%
3-gram word graph (m=3), within-word HMM word ends: 497, LM-states: 299, WER: 9.47%				
-	712.5	3.90	2.02%	9.47%
150k	176.1	2.64	2.11%	9.47%
100k	60.3	2.19	2.29%	9.47%
50k	13.5	1.82	3.60%	9.47%
20k	4.6	1.49	5.88%	9.47%

Table 3: 64K NAB task: properties of m-gram constrained word graphs and rescoring results for trigram LM.

7. REFERENCES

[1] X.L. Aubert, H. Ney, 'Large Vocabulary Continuous Speech Recognition Using Word Graphs', *Proc. of ICASSP, Detroit,* 1995, pp. 49–52.

[2] X.L. Aubert, 'One Pass Cross Word Decoding for Large Vocabularies Based on a Lexical Tree Search Organization', *Proc. of Eurospeech, Budapest,* 1999, pp. 1559–1562.

[3] D. Klakow, 'Log-Linear Interpolation of Language Models', *Proc. of ICSLP, Sidney,* 1998, pp. 1695–1698.

[4] Ch. Neukirchen, X.L. Aubert, H. Dolfing 'Extending the Generation of Word Graphs for a Cross-Word m-Gram Decoder', *Proc. of ICSLP, Bejing,* 2000, IV pp. 302–305.

[5] J.J. Odell, 'The Use of Context in Large Vocabulary Continuous Speech Recognition', *PhD Thesis, Engineering Department, Cambridge University,* 1995.

[6] M. Oerder, H. Ney, 'Word Graphs: An Efficient Interface between Continuous Speech Recognition and Language Understanding', *Proc. of ICASSP, Minneapolis,* 1993, pp. 119–122.

[7] H. Ney, X.L. Aubert, 'A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition', *Proc. of ICSLP, Yokohama,* 1994, pp. 1355–1358.

[8] H. Ney, S. Ortmanns, I. Lindam, 'Extensions to the Word Graph Method for Large Vocabulary Continuous Speech Recognition', *Proc. of ICASSP, Munich,* 1997, pp. 1791–1794.

[9] R. Schwartz, S. Austin, 'A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses', *Proc. of ICASSP, Toronto,* 1991, pp. 701–704.