Direct Observation of Pruning Errors (DOPE): A Search Analysis Tool

V. Steinbiss, M. Sundermeyer, and H. Ney

Chair of Computer Science 6, RWTH Aachen University

{steinbiss, sundermeyer, ney}@cs.rwth-aachen.de

Abstract

The search for the optimal word sequence can be performed efficiently even in a speech recognizer with a very large vocabulary and complex models. This is achieved using pruning methods with empirically chosen parameters and the willingness to accept a certain amount of pruning errors. Quite unsatisfying though, it is state-of-the-art that such pruning errors are not directly detected but, instead, indirect consequences of them, providing only a rough picture of what happens during search. With the tool *Direct Observation of Pruning Errors* (DOPE), described in this paper, pruning errors are detected on the state hypothesis level, which is a very fine level of granulation, several orders of magnitude finer than the sentence level. This allows much more exact analyses, including the analysis of pruning methods, or the effects of pruning parameters.

1. Introduction

Given a speech recognizer together with its knowledge sources—the pronunciation dictionary, the acoustic model and the language model—the task generally referred to as 'search' is to find the optimal word sequence. To achieve this goal using Viterbi decoding, the optimal state sequence is determined, giving rise to the optimal word sequence. It should be stressed that the search is just this optimization problem, such that different search procedures should always result in exactly the same (state and thus) word sequences with exactly the same probability values. If, due to pruning, a search procedure finds an optimal state sequence different from the one found by a full search, a (canonical) pruning error has occured.

Note that we say '*the* optimal' sequence as if it were unique—as usual, ties (identical scores) make the formulation clumsy, but are no real problem (they can be arbitrarily broken), so for the sake of a simpler presentation and without loss of generality, we completely ignore ties throughout this paper.

The basic ingredient of one-stage time-synchronous integrated beam search is to multiply out the available knowledge sources into a (possibly very large) state space and to unfold it over time (integrated search), to seek the optimal state sequence through this state space, working through time, frame by frame, from a start to an end time frame. Taking advantage of 'Bellman's optimality principle' and a process called 'recombination' [1] which only keeps track of state hypotheses that could in principle be part of an optimal path—and which discards all others—the computational complexity is only linear in time. In addition to this exact process that still guarantees finding the optimum, there is a heuristic process called 'pruning' which is motivated by the fact that, per time frame, the potential state hypothesis space is typically huge while the number of state hypotheses that have a realistic chance to be part of the optimal path is rather limited (in the range of thousands per time frame). By pruning (removing) hypotheses that are much worse than the optimal partial hypothesis up to this time frame and thus are unlikely to be part of the globally best path, the global optimum can still be found given that pruning was not too tight. There exist several simple and strong pruning methods saving, in a typical recognition system, several orders of magnitude of search effort. In contrast to recombination, pruning can result in losing the optimal path, i. e. in not finding the global optimum. Pruning errors can cause recognition errors but do not need to.

2. Concepts of a 'Pruning Error'

It seems obvious what a pruning error is. Let us from now on restrict to the state hypothesis rather than the word level. This choice provides finer granularity, and anyway, any pruning error on the word level comes from a pruning error on the state level. As a possible definition, a pruning error occurs when

1. Pruning a state hypothesis that would have resulted in the globally optimal path ('globally optimal' means at the end of the speech input, after exploitation of all available knowledge sources, in a full search).

However, other variants are also compelling. Try these:

2. Pruning a state (or word) hypothesis that results in an additional recognition error.
3. Pruning a state hypothesis that would have resulted in the globally optimal state hypothesis sequence through the actually *spoken* word sequence (see section 2).

It might be worth while reflecting why neither of the definitions implies the other.

(1) is clearly the 'right' and academically appealing definition—however, at least conceptually it has the drawback that it requires an exhaustive search that is in general practically not feasible. Even worse, its practical relevance is limited as we might not even care if one incorrect hypothesis is replaced by another incorrect one.

(2) describes the type of events that are to be avoided in a working system—however, this is conceptually immature as there can be 'helpful' pruning errors, and (2) is no more separated from the knowledge sources 'acoustic model' and 'language model'.

(3) is the definition used in this paper: It focuses on the only practically relevant mal-function of pruning, where the originally spoken sentence is excluded from competing with other hypotheses due to pruning.

2.1. Identifying Pruning Errors of type (1) and (2)

A pruning error of type (2) can be detected with a brute force approach: Run a series of experiments with different search methods or pruning parameters and compare the errors or the error rates, see e. g. [2]. The method makes high demands on computing resources while still not achieving a full search, and

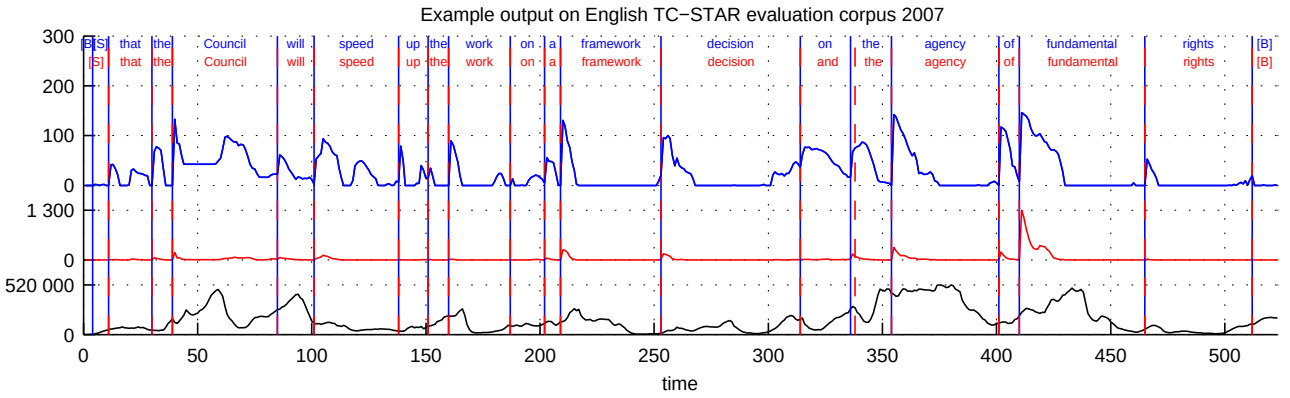

Figure 1: Example output of DOPE showing from top to bottom: (1) the transcribed word sequence, (2) the recognized word sequence, (3) the relative score of the spoken state hypothesis, (4) the number of state hypotheses with a score less or equal than the score of the spoken state hypothesis (ranging up to 1 300), and (5) the number of state hypothesis before pruning (ranging up to 520 000).

unfortunately also not giving further indication about the exact nature of the pruning error. It provides an indication of type (1) errors, but it will only uncover a subset of them.

Another method is based on a forced alignment of the known spoken sentence. With a decoding run on an HMM that allows exactly the constrained recognition of this sentence (plus optional non-speech sounds or silence), using exactly the same acoustic and language model scores, the resulting score S_{forced} can be compared with the score S_{free} of the optimal word sequence derived from a free (normal) recognition. (Scores are defined as negative log-probabilities, relative to the current best score of a given time frame.) If the criterion is met that S_{forced} is better than S_{free}, this proves that the search procedure has not found the optimal word sequence, i. e. a pruning error of type (1) has occurred. This criterion is obviously sufficient but not necessary. We would assume that it works better for shorter than for longer recordings and better for lower than higher word error rates.

3. The Concept of DOPE

The DOPE analysis tool follows a simple principle: For each time frame, it is checked whether, after pruning, the actually 'spoken' state hypothesis (for a definition see below) is still part of the active search space or whether it has been pruned (either at this point in time or earlier). Here are the details:

Given that the spoken word sequence is known (supervised situation), a Viterbi time alignment is performed (forced alignment), and the optimal state sequence is determined. For every time frame, the identity of the state as well as that of the history (the sequence of preceding words) is kept. Let us introduce the naming conventions *spoken state* and *spoken history*. Later on in the process of recognition, these will allow to identify the matching state hypothesis in the search space—in our system, for each time frame the HMM state hypothesis in a phonetic tree copy indexed with the language model context. Note that keeping the scores can be interesting to check the software but is conceptually superfluous.

The implementation of DOPE can be split up into three parts:

1. Knowing the spoken word sequence, determine the *spoken state* sequence with a forced alignment.
2. During decoding, for each time frame t check whether the *spoken state* hypothesis is active or pruned.
3. Do the statistics.

The implementation is straight-forward. For the connection (1) $\rightarrow$ (2), it is required to match the related but typically somewhat different data structures in training and decoding.

From now on, we focus on pruning errors of type (3):

Definition. When a state hypothesis—i. e. a pair (time frame, HMM state)—in the globally optimal state sequence through the actually spoken word sequence is pruned, this is a pruning error.

4. Experiments

Fig. 1 shows an example output of the DOPE tool. For its evaluation, we ran experiments using an MFCC-based English system that was trained on European Parliament Plenary Session (EPPS) data. Using system combination, it was one of four systems by RWTH that successfully took part in the TC-STAR evaluation campaign [3]. For simplicity, here we limit ourselves to the first recognition pass: Except for VTLN, we did not apply further adaptation techniques. Table 1 gives some statistics regarding the experimental setup.

EPPS English System on evaluation corpus 2007	
Audio	2.9 hrs
# running words	27 000
WER	14.7 %
OOV	0.0 %
# words in the lexicon	53 300
# n-grams, $n = 1, \ldots, 4$	7 400 000

Table 1: Key figures of the English EPPS system and the corresponding evaluation corpus; the full system attains a WER of 10.6 %. For our experiments, we used a lexicon having an OOV rate of 0 %, the original OOV rate being 0.1 %. The LM includes n-grams of an order of up to $n = 4$, during recognition we used bigram look-ahead.

4.1. Search error analysis

We ran several recognition passes with different values for the acoustic pruning threshold and measured the Word Error Rate (WER). One of the principal problems of this classical method can be seen in Fig. 2: On the left side the number of word errors is shown as a function of the acoustic pruning threshold (all other pruning parameters are kept fixed). Starting with high values, the WER remains constant, whereas for small values the number of word errors is very sensitive to small changes in the pruning parameter.

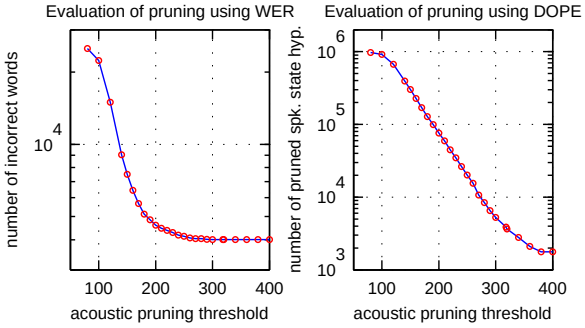

Figure 2: Errors due to pruning as a function of the acoustic pruning threshold. To minimize interference between different pruning strategies, we switched off histogram pruning for this experiment.

Furthermore, the WER method tends to produce slightly noisy data such that e. g. increasing the search space does not necessarily improve the WER: In our data, when relaxing the pruning threshold from 318 to 320 this resulted in increasing the number of word errors from 4 008 to 4 014.

On the right we can see the result of DOPE analyzing search errors on a state level. Over the whole parameter range we observe a noticable change in the number of pruned spoken state hypotheses so that pruning errors can now be detected which are hidden to the WER approach. No noise is visible.

At a WER of 14.7 % (4 011 incorrect words), the score of only 0.2 % of the spoken state hypotheses (1 763 state hypotheses) exceeds the pruning threshold. So we can derive that an increase in the pruning parameters hardly will improve the recognition result, and almost all word errors are due to modelling inaccurracies. In contrast, from the WER data it is not clear whether an increase in the pruning parameters would still improve the WER.

4.2. Inspection of the search effort

A detailed profiling of the RWTH decoder shows that about 60 % of the CPU effort for a recognition pass results from the evaluation of the acoustic model. The remaining CPU time is equally distributed between language model (LM) lookups, the LM look-ahead (LM-LA) computation, and the state expansion. As a result, more than 70 % of the runtime directly depends on the number of state hypotheses that are active during search. We therefore analyze the state space in more detail.

Fig. 3 (a) shows that the number of state hypotheses grows strongly with increasing score. The converse holds true for spoken state hypotheses, see Fig. 3 (b). Most search effort is spent on evaluating state hypotheses that are unlikely to eventually become part of the optimum state sequence. Regarding the size of the search space, on average we have 21 177.1 hypotheses left after applying a tight pruning that still preserves the optimum WER. At the same time, on average only 637.6 hypotheses have a score less or equal than the score of the spoken state hypothesis, which is a lower bound for the minimum search space size.

In [4], the authors state that most search effort is spent in the first two phoneme generations of the state tree. This is confirmed by our experiments: Fig. 3 (c) indicates that about 60 % of all state hypotheses (before pruning) can be found in the first two phoneme generations of the state tree, and 99 % of them lie within the first seven phoneme generations (we used three states per phoneme).

For spoken state hypotheses, Fig. 3 (d), the recognition effort reflected by the state frequencies directly follows the word lengths observed in the recognition corpus. This result is only interferred by the choice of HMM transitions of the search procedure.

4.3. Inspection of pruning methods

We now examine the performance of our DOPE tool by investigating whether it finds experimental evidence for the properties that several state-of-the-art pruning techniques rely on.

The *acoustic pruning* [1] obviously makes use of the fact that scores of spoken states tend to be much smaller than in the average case, as observed in the previous section, Fig. 3 (b).

In practice, parameter values for *LM pruning* [1] can be much smaller than those for the acoustic pruning without increasing the WER. In principle, LM pruning can be regarded as applying acoustic pruning to the set of word end states. We find that the score distribution for this subset of state hypotheses does not look different from the one we saw for all hypotheses in Fig. 3 (a).

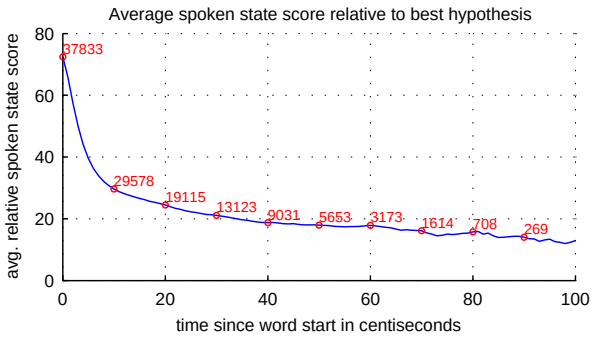

Figure 4: Score of spoken state relative to the current optimum, averaged over all words; as the number of observations (numbers at the markers) decreases over time, values for time > 100 tend to be noisy (less than 100 observations).

Instead, in Fig. 4 we see that scores of spoken states of a word continuously decrease over time—as opposed to the average case where scores more or less remain constant. As scores of spoken states at word end times are smaller, we suppose that the LM pruning beam can be tighter. In any case, a tighter pruning at word ends will limit the number of newly started trees, which greatly reduces the search effort that is dominated by the first generations of the prefix tree. At last, a reduced number of trees will also minimize the overhead for the computation of LM look-ahead values.

The *LM look-ahead* [1] tries to incorporate the LM information as early as possible. In Fig. 4 we have already seen that

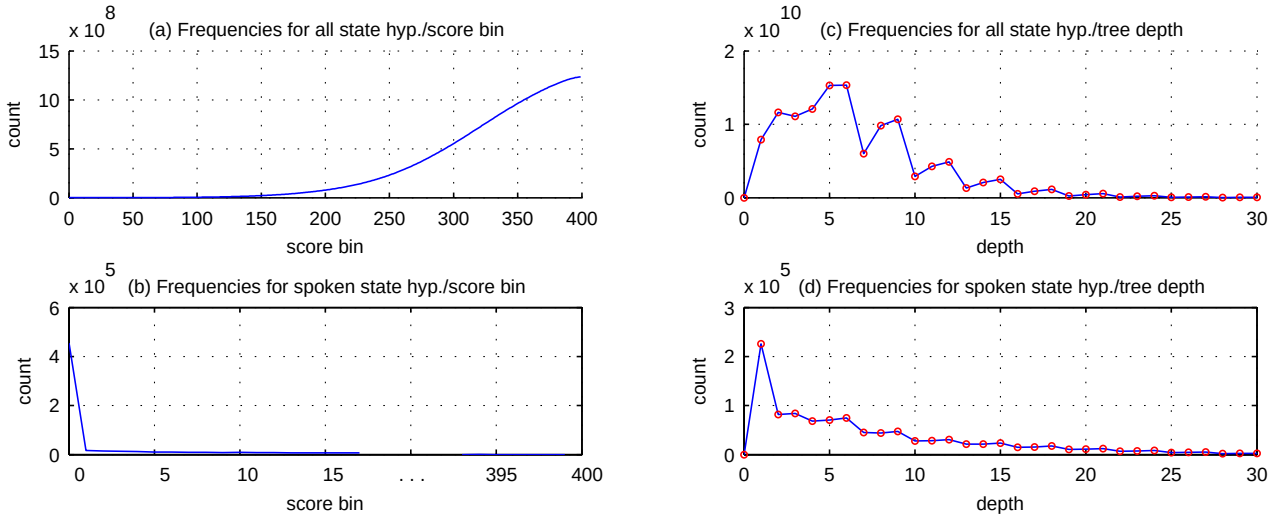

Figure 3: Frequency counts before pruning for all active/spoken state hypotheses as a function of depth/score bin; for integer b, a score bin b is defined as $b := \lfloor s \rfloor$, $0 \leq b < 400$, where s is the score of a state hypothesis.

at word start times the score of spoken state hypotheses shows a large peak. This is caused by the LM look-ahead adding at word start times a score for the most likely ending word.

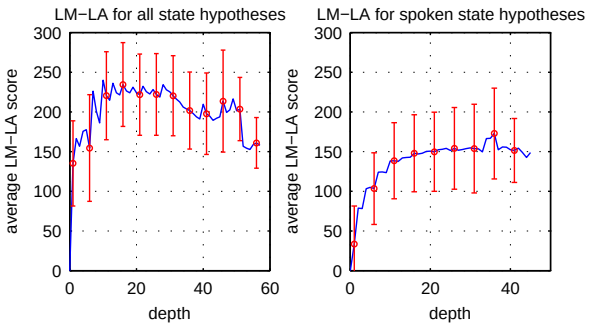

Figure 5: Average LM look-ahead as a function of depth in the lexical prefix tree, error bars indicating $+/-$ standard deviation

We assume that these peaks can pose problems because during time-synchronous search, word start hypotheses including the LM-LA peak are compared with word end hypotheses where this information has not been incorporated yet.

Nevertheless, LM-LA leads to a considerable improvement. Without any look-ahead, the peak would occur at word end times, and the value of the peak would be equal to the LM-LA at word end times. (In fact, as we use a 4-gram LM whereas the LM-LA is based on a bigram LM, both values might slightly differ.) Using LM-LA we find that the peak, now occurring at word start times, is much smaller, see Fig. 5. Second, we observe that the curve for spoken state hypotheses is even smoother, further reducing the LM peak.

5. Conclusions and Future Work

The direct observation of pruning errors (DOPE) is a tool that observes the effects of pruning strategies and parameters on the state hypothesis level and thus provides much deeper insight than other methods. The presented experiments show the validity of the approach. Furthermore, we can conclude that current pruning methods are not at their limits and may still be improved in terms of the resulting search space size.

We have presented DOPE in the context of time-synchronous beam search. However, it should be noted that the principle of DOPE is rather general and can be used as well for other search procedures such as stack decoding or transducer-based search [4] and also in different areas such as statistical machine translation.

Further investigations are possible that extend beyond the scope of this paper, for example the location of errors (short words, word boundaries, silence, etc.) or the calibration of pruning methods having large sets of free parameters.

Acknowledgments: This work was realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation. — The idea dates back to 1998 when one of the authors (VS) worked at Philips Research Labs, Aachen. First experiments were done by Hauke Schramm, now at Fachhochschule Kiel.

6. References

[1] H. Ney and S. Ortmanns, "Dynamic programming search for continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 64–83, 1999.

[2] J. Pylkkönen, "New pruning criteria for efficient decoding," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[3] J. Lööf, C. Gollan, S. Hahn, G. Heigold, B. Hoffmeister, C. Plahl, D. Rybach, R. Schlüter, and H. Ney, "The RWTH 2007 TC-STAR evaluation system for European English and Spanish," in *Proc. Int. Conf. on Speech Communication and Technology, Antwerp, Belgium*, 2007, pp. 2145–2148.

[4] X. Aubert, "An overview of decoding techniques for large vocabulary continuous speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 89–114, 2002.