A COMPARISON OF TWO LVR SEARCH OPTIMIZATION TECHNIQUES

Stephan Kanthak *, *Hermann Ney*

Lehrstuhl für Informatik VI,
Computer Science Department,
RWTH Aachen – University of Technology,
52056 Aachen, Germany
{kanthak, ney}@informatik.rwth-aachen.de

Michael Riley, Mehryar Mohri

AT&T – Research
Speech Algorithms Research Department
180 Park Avenue
Florham Park, NJ 07932-0971, USA
{riley,mohri}@research.att.com

ABSTRACT

This paper presents a detailed comparison between two search optimization techniques for large vocabulary speech recognition – one based on word-conditioned tree search (WCTS) and one based on weighted finite-state transducers (WFSTs). Existing North American Business News systems from RWTH and AT&T representing each of the two approaches, were modified to remove variations in model data and acoustic likelihood computation. An experimental comparison showed that the WFST-based system explored fewer search states and had less runtime overhead than the WCTS-based system for a given word error rate. This is attributed to differences in the pre-compilation, degree of non-determinism, and path weight distribution in the respective search graphs.

1. INTRODUCTION

Over the past 15 years many different search algorithms have been developed for large-vocabulary speech recognition. [1] gives an extensive overview and summary of current popular approaches for representing and traversing the search space in large vocabulary speech recognition systems. In this paper, we compare in detail two of these approaches – one based on *word-conditioned tree search* and one based on *weighted finite-state transducers*.

The paper is organized as follows. In Sections 2 and 3, we give a short overview of each of these methods. In Section 4, we compare the two approaches in detail, discussing technical advantages and disadvantages and concluding with a detailed experimental comparison.

2. WORD-CONDITIONED TREE SEARCH

In the word-conditioned tree search (WCTS) approach, portions of the phonetic search space are pre-compiled into lexical trees. In this way, words that share the same prefix can share computation during recognition. The language model weights are introduced in an on-demand copy of the tree conditioned by word context and are distributed along the branches of the tree to improve pruning. This technique is called language model look-ahead. A detailed description of the WCTS approach, can be found in [2] and a description of the necessary extensions to use cross-word context-dependent phoneme models can be found in [3].

3. WEIGHTED FINITE-STATE TRANSDUCERS

In this approach, all components used in the search stage of the ASR system – language model, pronunciation dictionary, phonetic context-dependency, HMM model – are represented as weighted finite-state transducers (WFSTs) [4, 5]. These individual models are then combined and optimized using the general weighted finite-state operations of composition, determinization, minimization and weight-re-distribution ("pushing") [6, 5, 7]. The purpose of these steps is to create a single optimized transducer that maps directly sequences of HMM-state-level distributions to sequences of words. Since the search transducer is built statically, these optimizations can be performed entirely off-line. The composition, determinization and minimization algorithms, a non-deterministic language model back-off representation [8] remove redundancy and minimize the size of the recognition transducer. Language model shrinking [9] allows control of the size and accuracy trade-off even for very large vocabulary tasks.

4. COMPARISON OF APPROACHES

All experimental comparisons were carried out on the DARPA North American Business News (NAB) 1995 evaluation corpus. There exist recognition systems that have been optimized for this particular task in the past [3] [2]. For our comparisons, we use a WCTS-based system developed at RWTH and a WFST-system developed at AT&T.

* The author performed the work as a summer student at AT&T with the supervision of Michael Riley.

A direct comparison of the efficiency of the search space representation of the two systems is difficult due to differences in the decoder implementations and model parameterizations. Differences between the two complete systems are compiled in Table 1. The best single-pass word error rates of each system for this task is shown in Table 2 as well as some additional combinations of the models as the two decoder implementations allowed.

For experimental comparison we used the AT&T language model and the RWTH acoustic model providing a baseline word error rate of 12.7%.

Table 1. Comparison of the systems' parameters.

	AT&T	RWTH
LDA	no	yes
dim. of features	39	33
# mixtures	$\approx 7,200$	$\approx 5,000$
# densities	$\approx 85,000$	$\approx 300,000$
variance	diagonal, per density	diagonal, global
HMM topology	3 states	6 states
duration model	gamma distribution	3 segments, skip / forward / loop
vocabulary	$40,000$	$64,000$
pron. variants	$\approx 2,000$	$\approx 2,000$
language model	trigram	trigram
perplexity	134	146

Table 2. Comparison of the systems optimized for word error rate (AT&T LM with shrinking factor of 40).

Decoder	Acoustic Model	Language Model	WER [%]
RWTH	RWTH	RWTH 64K	10.1
	RWTH	AT&T 40K	12.7
AT&T	AT&T	AT&T 40K	14.7
	RWTH	AT&T 40K	12.8

4.1. Technical Comparison

Although the two search approaches are based on almost the same statistical models, they differ in their design paradigms. The WFST approach statically pre-compiles and optimizes the whole network based on all knowledge sources. The pre-compiling steps usually take large amounts of memory for LVCSR tasks (e.g. peak 4GB for shrink 40 in Table 3). In contrast, the WCTS approach pre-compiles only the lexicon and acoustic model into a prefix tree and performs on-the-fly composition with the language model during recognition of an utterance. Due to additional steps during recognition we expect that the effective runtime of the WFST approach is smaller.

In the WCTS approach, the lexical prefix tree structure only eliminates non-determinism for words starting with the same phonetic pronunciation. Minimization of the WFST approach has no counterpart in the WCTS approach, because tails of words are not optimized at all in the WCTS approach. In the WCTS approach the recombination at the word level has to be coded explicitly whereas in the WFST approach the language model recombination is part of the construction of transducer G. The effect of the language model look-ahead is close to that of weight pushing used for WFSTs, but it is restricted to a single tree copy and limited to sequences preceeding word boundaries.

In order to pre-compile a single recognition transducer, the WFST construction exploits the sparseness of the knowledge sources. Exploiting the sparseness of the language model includes, in particular, taking advantage of the specific back-off structure of the model. As a result of this, the size of the LM automaton (which has W^2 states and W^3 arcs in theory for a full W words trigram LM) is almost proportional to the number of modeled events, although it is no longer deterministic in that case. Exploiting the backing-off structure of the LM is also possible for the WCTS approach by simply creating tree copies only for distinct backing-off histories [10], but this has not been used in this comparison. It should be noted that the WCTS approach may use any arbitrary language model (in particular language models that are constructed dynamically).

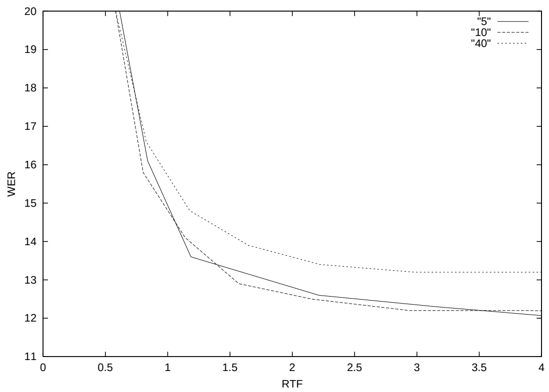

Fig. 1. Effect of different shrinking factors on the ratio between WER and RTF. The ratio with shrinking factors of 5 and 10 is almost equal. For shrinking with 40 the WER increases by about 10% relative. Note the sizes of the final transducers for different shrinking factors (see Table 3).

4.1.1. Approximations

Even when exploiting the backing-off structure, the LM transducer G may be large resulting in a large $HCLG$ recognition transducer causing the recognizer to use a lot of computational resources. To reduce these requirements, G may be shrunk to a smaller size, i.e. with fewer events modeled [9]. This approximation may affect the recognition accuracy. Figure 1 shows that for this task the effect on the word error rate (WER) is quite small for moderate shrinking factors. Figure 1 also shows that even with the large difference in size of the final transducers (see Table 3) the relation between real-time factor (RTF) and WER does not change much. In the case of a shrinking factor of 40 the WER increases by 10% from 12.0% to 13.1% asymptotically, but for shrinking factors of 5 and 10 there is no noticeable difference. We used the integrated transducer based on the shrink 40 language model in all following experiments.

Table 3. Sizes of final transducers for different language model shrinking factors (sequences of HMM states are already factored).

shrinking factor	states	arcs
5	8,741,365	19,063,196
10	6,666,178	14,294,667
40	2,749,050	5,862,012

The language model (LM) look-ahead as implemented in the RWTH decoder is based on unigram or bigram probabilities even when using a trigram LM for recognition. Experiments with the WCTS approach have shown that a trigram LM look-ahead degrades recognition speed compared to an approximative bigram LM look-ahead [11]. The trigram LM look-ahead calculates partial LM probabilities for every tree copy while for the bigram LM look-ahead this is only necessary for trees with different predecessor words.

The pushing algorithm of the WFST framework uses the full language model probabilities. The underlying semiring for pushing has a strong influence on the effectiveness of pushing with respect to recognition speed. As shown in [7] the log (“sum of probabilities”) semiring significantly outperforms the tropical (”maximum of probabilities”) semiring. The LM look-ahead as used in the WCTS approach is based on the tropical semiring. However, in contrast to the results stated in [7] the LM look-ahead improves pruning and recognition speed, because partial LM probabilities are not pushed beyond word boundaries.

4.1.2. Implementation

The WFST approach is a typical token passing algorithm without any context information attached to the tokens. Therefore the decoder can be written as a simple loop over a set of active states and another one over all time frames to be recognized.

In contrast, the WCTS approach has its loops grouped according to the outermost contexts (language model histories and arcs in the lexicon) and therefore needs at least two loops over the set of active search states for each time frame. In addition, when using cross-word context dependent phoneme models, phonetic arcs at the root and the leaves of the prefix tree need special treatment (see [3] for details).

Altogether, the implementation of the AT&T decoder for WFSTs is simpler compared to the RWTH WCTS decoder which positively affects processor cache usage.

4.1.3. Flexibility

The WFST approach is a general framework and expresses all speech recognition knowledge sources with the same transducer representation. Instead of the many specific optimizations developed for the WCTS approach their counterparts for the WFST even do a better job, because they apply to the whole network, rather than to a local part of it. More formally, the WCTS approach can be expressed by means of the WFST framework when using on-the-fly composition.

Speech recognition using the WFST framework is also not restricted to a fixed number of four transducers to build the final network.

4.2. Experimental Comparison

Experiments were carried out using the NAB’95 evaluation corpus. We used the RWTH acoustic model and the AT&T language model for both systems in the direct comparison. The RWTH acoustic model likelihood computation code was shared between the two systems to eliminate that as a source of variation. For all experiments, we used an 850 MHz Pentium III processor running Linux.

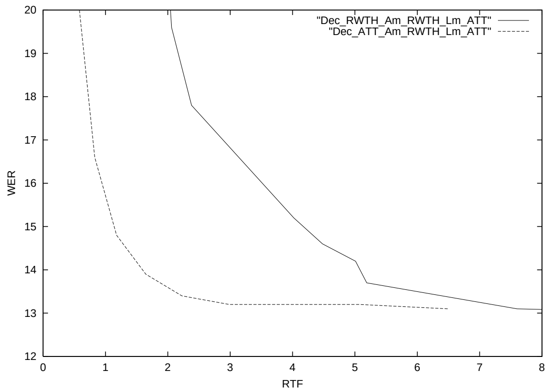

Fig. 2. Word error rate versus real-time factor for the same models using the two different approaches. The WCTS approach is about 3 times slower than the WFST approach.

Figure 2 compares the word error rate versus real-time factor for both search implementations. The WCTS approach is about 3 times slower than the WFST approach.
During recognition, the WFST approach only needs system memory for the pre-compiled final transducer $HCLG$, structures in order to calculate acoustic model emission probabilities and some minor amount of memory for dynamically generated state hypotheses. The WCTS approach additionally uses a huge amount of memory in order to cache the LM look-ahead probabilities. For the runs on the NAB corpus, the AT&T decoder using the WFST approach consumed about 400MB of memory while the RWTH decoder using the WCTS approach consumed about 700MB of memory.

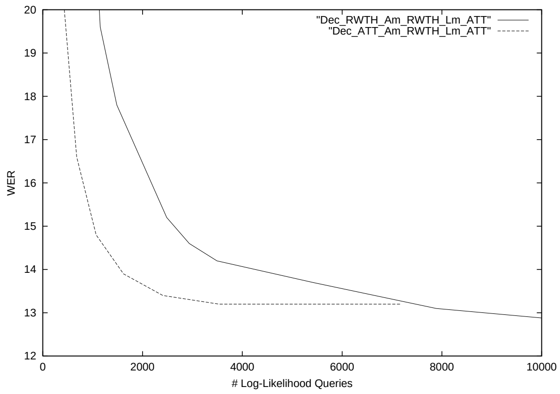

Fig. 3. Word error rate versus number of state emission probability queries. Here, WCTS is only about a factor of 2 away from WFST which emphasizes the influence of the additional run-time overhead (see also Figure 4).

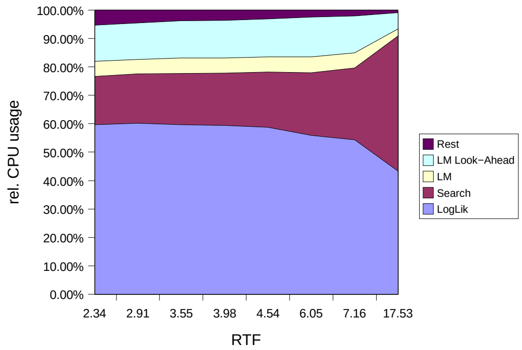

Fig. 4. Relative CPU usage of the different components for the WCTS approach. Obviously, likelihood calculation takes most of the CPU time, but token management (search) comes next, before language model look-ahead.

5. DISCUSSION

We have shown that the WCTS approach is about 3 times slower than the WFST approach. We believe that this is mostly due to the conceptually less effective LM look-ahead, necessary overhead for dynamic expansion and the more complex implementation. Figure 3 shows that the number of log-likelihood queries (equivalent to the number of active states after pruning) is only about a factor of 2 higher than those of the WFST approach. This emphasizes the influence of the additional run-time overhead.

The curves for the WCTS approach are not as smooth as the one for the WFST approach: there is only one pruning parameter necessary for the WFST approach, while there are three for the WCTS approach (separate beam pruning thresholds for fan-out and all other arcs and another one for hypotheses after application of language model probabilities before new trees are started).

6. REFERENCES

[1] X. Aubert, "A Brief Overview of Decoding Techniques for Large Vocabulary Continuous Speech Recognition," in *ISCA ITRW ASR 2000 Automatic Speech Recognition: Challenges for the New Millenium*, Paris, France, Sep. 2000, pp. 91 – 96.

[2] H. Ney, L. Welling, S. Ortmanns, K. Beulen, and F. Wessel, "The RWTH Large Vocabulary Continuous Speech Recognition System," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Seattle, WA, May 1998, pp. 853 – 856.

[3] S. Kanthak, A. Sixtus, S. Molau, R. Schlüter, and H. Ney, "Fast Search for Large Vocabulary Speech Recognition," in *Verbmobil: Foundations of Speech-to-Speech Translation*, Wolfgang Wahlster, Ed., pp. 63 – 78. Springer, Berlin, 2000.

[4] F. Pereira and M. Riley, *Finite State Language Processing*, chapter Speech Recognition by Composition of Weighted Finite Automata, The MIT Press, 1997.

[5] M. Mohri, F. Pereira, and M. Riley, "Weighted Finite-State Transducers In Speech Recognition," in *ISCA ITRW ASR 2000 Automatic Speech Recognition: Challenges for the New Millenium*, Paris, France, Sep. 2000, pp. 97 – 106.

[6] M. Mohri and M. Riley, "Integrated Context-Dependent Networks in Very Large Vocabulary Speech Recognition," in *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech '99)*, Budapest, Hungary, 1999.

[7] M. Mohri and M. Riley, "A Weight Pushing Algorithm for Large Vocabulary Speech Recognition," in *European Conf. on Speech Communication and Technology*, Aalborg, Denmark, Sep. 2001, pp. 1603 – 1606.

[8] G. Riccardi, E. Bocchieri, and R. Pieraccini, "Non-Deterministic Stochastic Language Models for Speech Recognition," in *Proc. ICASSP*. 1995, vol. 1, pp. 237–240, IEEE.

[9] K. Seymore and R. Rosenfeld, "Scalable Backoff Language Models," in *Proceedings of ICSLP*, Philadelphia, Pennsylvania, 1996.

[10] F. Brugnara and M. Cettolo, "Improvements in Tree-Based Language Model Representation," in *European Conf. on Speech Communication and Technology*, pp. 1797 – 1800. Madrid, Spain, Sep. 1995.

[11] S. Ortmanns, *Effiziente Suchverfahren zur Erkennung kontinuierlich gesprochener Sprache*, Ph.D. thesis, RWTH Aachen, Nov. 1998.