SHRINKING LANGUAGE MODELS BY ROBUST APPROXIMATION

Adam L. Buchsbaum[*] *Raffaele Giancarlo*[†] *Jeffery R. Westbrook*[‡]

ABSTRACT

We study the problem of reducing the size of a language model while preserving recognition performance (accuracy and speed). A successful approach has been to represent language models by weighted finite-state automata (WFAs). Analogues of classical automata determinization and minimization algorithms then provide a general method to produce smaller but equivalent WFAs. We extend this approach by introducing the notion of *approximate determinization*. We provide an algorithm that, when applied to language models for the North American Business task, achieves 25–35% size reduction compared to previous techniques, with negligible effects on recognition time and accuracy.

1. INTRODUCTION

An important goal of language model engineering is to produce small language models that guarantee fast and accurate automatic speech recognition (ASR). In practice we see tradeoffs: e.g., in size vs. accuracy and in accuracy vs. speed. There has been recent progress, however, on automatic methods for reducing the sizes of language models while preserving overall recognition performance (accuracy and speed). Lacouture and De Mori [5] and Kenny et al. [4] discuss the size reduction of lexical trees in order to speed ASR systems, but they deal only with unweighted trees. Pereira et al. [7,8] discuss the use of weighted finite-state automata (WFAs) to model human language in ASR systems. An advantage to this approach is that classical techniques for automata determinization and minimization can be extended to produce WFAs that are smaller yet formally equivalent to their inputs; i.e., each input string is assigned the same cost in both the input and the reduced models. Mohri [6] develops these ideas and reports significant size reductions from their application to ASR word lattices.

[*] AT&T Labs, 180 Park Ave., Florham Park, NJ 07932, U.S.A., alb@research.att.com

[†] Dipartimento di Matematica ed Applicazioni, Universitá di Palermo, Via Archirafi 34, 90123 Palermo, Italy, raffaele@altair.math.unipa.it. Work supported by AT&T Labs.

[‡] AT&T Labs, 180 Park Ave., Florham Park, NJ 07932, U.S.A., jeffw@research.att.com

We extend Mohri's approach by relaxing the requirement that the output model be formally equivalent to the input. We introduce a technique that we call *approximate determinization*, which produces a WFA that accepts precisely the same strings in a given language but only approximates their costs. Clearly, by using arbitrarily different costs (e.g., zero for all strings), we could produce trivially small language models that would perform poorly in ASR systems. Therefore, our algorithm accepts an approximation parameter that controls the tradeoff between output model size and recognition performance.

In experiments on NAB language models, we achieve 25-35% size reduction compared to Mohri's exact determinization and minimization algorithms, without affecting ASR time or accuracy. These results are invariant under different beam widths. We also show that our approximation algorithm preserves many of the n-best strings in the ASR output. Our technique is completely general and applicable to any language model.

Section 2 describes our algorithm. Section 3 reports experimental results showing size reduction and preservation of n-best strings. Section 4 reports experimental results showing size reduction and word accuracy. We conclude in Section 5.

1.1. Definitions

A *weighted finite automaton* (WFA) is a tuple $A = (Q, \bar{q}, \Lambda, \delta, Q_f)$ such that Q is the set of states, $\bar{q} \in Q$ is the initial state, Λ is the set of labels, $\delta \subseteq Q \times \Lambda \times \Re^+ \times Q$ is the set of transitions, and $Q_f \subseteq Q$ is the set of final states. A is also called a *lattice*. The *size* of A is $|A| = |Q| + |\delta|$. A *deterministic*, or *sequential*, WFA has at most one transition $t = (q_1, \sigma, \nu, q_2)$ for any pair (q_1, σ); a *nondeterministic* WFA can have multiple transitions on a pair (q_1, σ), differing in target state q_2.

Let $\vec{t} = (t_1, \ldots, t_\ell)$ be some sequence of transitions, such that $t_i = (q_{i-1}, \sigma_i, \nu_i, q_i)$; $\vec{t}$ *induces* string $w = \sigma_1 \cdots \sigma_\ell$. String w is *accepted by* $\vec{t}$ if $q_0 = \bar{q}$ and $q_\ell \in Q_f$; w is *accepted by* A if some $\vec{t}$ accepts w. Let $c(t_i) = \nu_i$ be the *weight* of t_i. Then the *weight of* $\vec{t}$ is $c(\vec{t}) = \sum_{i=1}^{\ell} c(t_i)$. Let $T(w)$ be the set of all sequences of transitions that accept string w. Then the *weight of* w is $c(w) = \min_{\vec{t} \in T(w)} c(\vec{t})$. The *lan-*

guage of A is the set of weighted strings accepted by A: $L(A) = \{(w, c(w)) : w \text{ is accepted by } A\}$.

When WFA's are used to represent language models, the weights are interpreted as negative log probabilities. The weight assigned to a string by the WFA is therefore the negative log probability of the string occurring in the input speech. All definitions and algorithms apply to WFAs over arbitrary semirings [1–3].

2. ALGORITHM

Mohri's automatic method for reducing the size of a language model is based on classical automata theory. First the input WFA, A, which is normally nondeterministic, is replaced by an equivalent deterministic WFA, A'. From A', an equivalent deterministic WFA, A'', of minimum size is computed. Most of the size reduction comes during the process of determinization. We first describe Mohri's algorithm for WFA determinization, which we call $\mathcal{D}$. We then present observations about its behavior that motivate our approximate determinization algorithm, $\tilde{\mathcal{D}}$.

Given WFA $A = (Q, \bar{q}, \Lambda, \delta, Q_f)$, Mohri generalizes the classical power-set construction to build deterministic WFA A' as follows. The start state of A' is $\{(\bar{q}, 0)\}$, which forms an initial queue P. While $P \neq \emptyset$, pop state $q = \{(q_1, r_1), \ldots, (q_n, r_n)\} \in 2^{Q \times \Re^+}$ from P. The r_i values encode path-length information, as follows. For each $\sigma \in \Lambda$, let $\{q'_1, \ldots, q'_m\}$ be the set of states reachable by σ-transitions out of all the q_i. For $1 \leq j \leq m$, let $\rho_j = \min_{1 \leq i \leq n; (q_i, \sigma, \nu, q'_j) \in \delta} \{r_i + \nu\}$ be the minimum of the weights of σ-transitions into q'_j from the q_i plus the respective r_i. Let $\rho = \min_{1 \leq j \leq m} \{\rho_j\}$. Let $q' = \{(q'_1, s_1), \ldots, (q'_m, s_m)\}$, where $s_j = \rho_j - \rho$ for $1 \leq j \leq m$. If q' is new, then it is pushed onto P. Transition (q, σ, ρ, q') is added to A'. This is the only σ-transition out of state q, so A' is deterministic.

Let $T_A(w)$ be the set of sequences of transitions in A that accept a string $w \in \Lambda^*$; let $\vec{t}_{A'}(w)$ be the (one) sequence of transitions in A' that accepts the same string. It can be shown that $c(\vec{t}_{A'}(w)) = \min_{\vec{t} \in T_A(w)} \{c(\vec{t})\}$, and thus $L(A') = L(A)$. Moreover, let $T_A(w, q)$ be the set of sequences of transitions in A from state $\bar{q}$ to state q that induce string w. Again, let $\vec{t}_{A'}(w)$ be the (one) sequence of transitions in A' that induces the same string; $\vec{t}_{A'}(w)$ ends at some state $\{(q_1, r_1), \ldots, (q_n, r_n)\}$ in A' such that some $q_i = q$. It can be shown that $c(\vec{t}_{A'}(w)) + r_i = \min_{\vec{t} \in T_A(w,q)} \{c(\vec{t})\}$. Thus each r_i is a *remainder* that encodes the difference between the weight of the shortest path to some state that induces w in A and the weight of the path inducing w in A'. At least one remainder in each state is thus zero.

Given the importance of weights in WFA determinization, we studied the effects of $\mathcal{D}$ on a few ATIS word lattices. We found that when tuples of states occurred multiple times, with different remainders, the remainders tended to be "clustered" in the following sense. Most corresponding remainders were equal; only one or two differed between the tuples, and the differences were small relative to magnitude.

This led to our *approximate determinization* algorithm, $\tilde{\mathcal{D}}$, which generalizes Mohri's algorithm as follows. In addition to a WFA A, $\tilde{\mathcal{D}}$ takes an *approximation parameter* ε as input. When constructing a state $q' = \{(q_1, r'_1), \ldots, (q_\ell, r'_\ell)\}$ in the determinized WFA A', if there exists some previously constructed state $q = \{(q_1, r_1), \ldots, (q_\ell, r_\ell)\}$ such that, for $1 \leq i \leq \ell$, $|r'_i - r_i| \leq \varepsilon \cdot \min\{r_i, r'_i\}$, we use q in place of q'. (If some r_i=0, the corresponding r'_i must also be zero.) A' is still deterministic and accepts exactly the same strings as A but may assign different weights to those strings.

Figure 1 gives pseudocode for $\tilde{\mathcal{D}}$. In the next two sections, we measure the quality of $\tilde{\mathcal{D}}$. We first consider how it preserves the n-best strings in word lattices. We then show how it affects recognition performance when applied to language models.

3. APPLICATION TO WORD LATTICES

In this section, we study the quality of $\tilde{\mathcal{D}}$ by measuring how well it preserves the n-best strings of word lattices. We applied $\tilde{\mathcal{D}}$ with various approximation parameters to 100 word lattices generated by the AT&T North American Business speech recognizer [9], using an ATIS grammar. For each lattice, we measured the resulting size reduction, and we determined the maximum n such that the n-best strings in the input appeared, in the same order, as the n-best strings in the output of $\tilde{\mathcal{D}}$. We averaged the results over the test set.

Figure 2 depicts the resulting size reduction. The x-axis shows different values of ε, and the y-axis compares sizes of the results with those of the input word lattices. Even small values of ε gave reduction factors better than .4 for states and .2 for transitions.

Figure 3 shows that even using large values of ε did not affect the best paths through the lattices. Here the y-axis shows the maximum n (up to $n = 20$) such that the n-best strings were preserved in order.

4. RECOGNITION PERFORMANCE

In this section, we study the quality of $\tilde{\mathcal{D}}$ by applying it, with different values of ε, to shrink the composed lexicon and grammar for the NAB task. We ran 300

1. Input: $A = (Q, \bar{q}, \Lambda, \delta, Q_f)$;
2. Input: Approximation parameter ε;
3. Let $Q', Q'_f, \delta' \leftarrow \emptyset$;
4. Let $P = \{\{(\bar{q}, 0)\}\}$;
5. While $P \neq \emptyset$ do
6. Remove any element $q = \{(q_1, r_1), \ldots, (q_n, r_n)\}$ from P;
7. $Q' \leftarrow Q' \cup \{q\}$;
8. For all $\sigma \in \Lambda$ do
9. Let $\{q'_1, \ldots, q'_m\} \leftarrow \bigcup_{1 \leq i \leq n} \{q' : \exists \nu, (q_i, \sigma, \nu, q') \in \delta\}$;
10. Comment: these are the states (up to multiplicity) reachable by σ-transitions out of all the q_i.
11. For $1 \leq j \leq m$ let $\rho_j \leftarrow \min_{1 \leq i \leq n; (q_i, \sigma, \nu, q'_j) \in \delta} (r_i + \nu)$;
12. Let $\rho = \min_{1 \leq j \leq m} \rho_j$;
13. For $1 \leq j \leq m$ let $s_j = \rho_j - \rho$;
14. Let $q' \leftarrow \{(q'_1, s_1), \ldots, (q'_m, s_m)\}$;
15. Let $T_1 \leftarrow \{\{(q''_1, x_1), \ldots, (q''_m, x_m)\} \in Q' \cup P \cup \{q\} : q''_i = q'_i,\ x_i \in \Re^+,\ 1 \leq i \leq m\}$;
16. Comment: T_1 is the set of tuples with states identical to those of q';
17. Let $T_2 \leftarrow \{\{(q''_1, x_1), \ldots, (q''_m, x_m)\} \in T_1 : |x_i - s_i| \leq \varepsilon \cdot \min\{s_i, x_i\}\}$;
18. If $T_2 = \emptyset$ then
19. $P \leftarrow P \cup \{q'\}$;
20. If $q'_i \in Q_f$ for any $1 \leq i \leq m$ then $Q'_f \leftarrow Q'_f \cup \{q'\}$;
21. Let $\delta' \leftarrow \delta' \cup \{(q, \sigma, \rho, q')\}$;
22. else
23. Pick q'' to be any element of T_2;
24. Let $\delta' \leftarrow \delta' \cup \{(q, \sigma, \rho, q'')\}$;
25. End If;
26. End For;
27. End While;
28. Output: $A' = (Q', (\bar{q}, 0), \Lambda, \delta', Q'_f)$.

Figure 1: Approximate determinization algorithm.

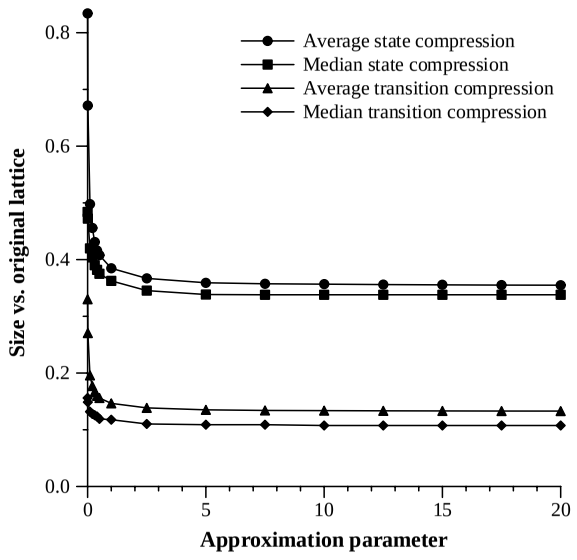

Figure 2: Size reduction vs. approximation.

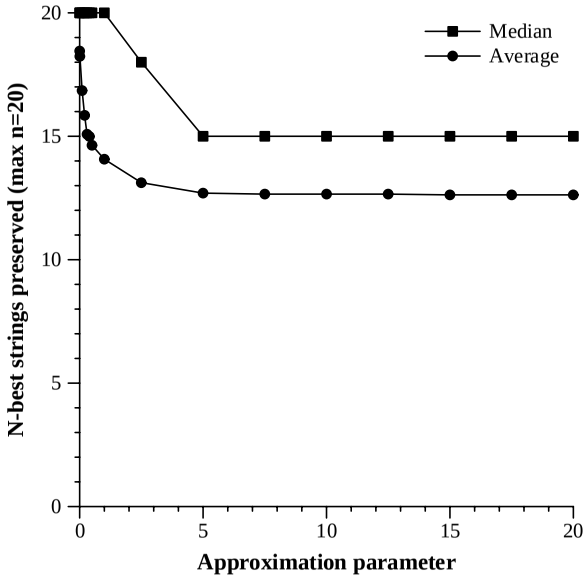

Figure 3: Preservation of best strings.

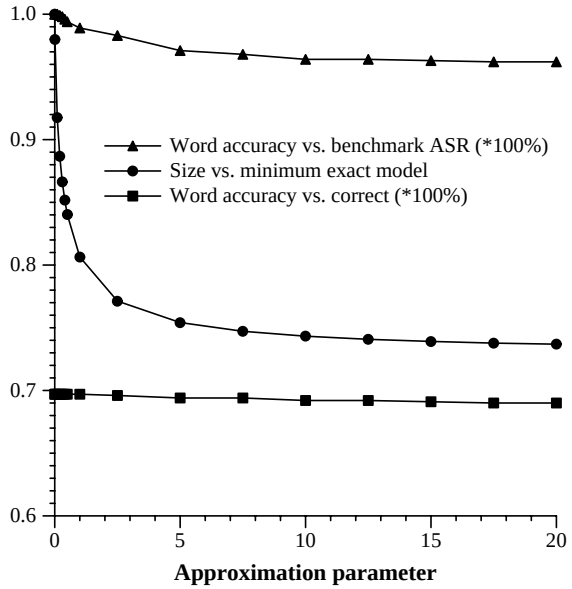

Figure 4: Recognition performance; small NAB models.

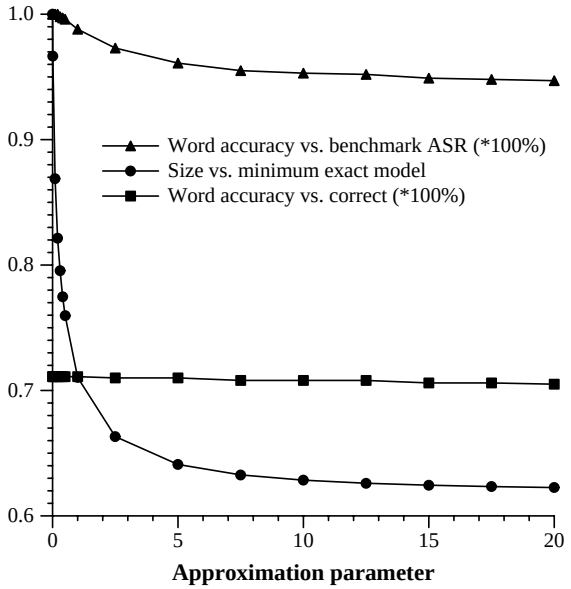

Figure 5: Recognition performance; large NAB models.

sentences using these models in the AT&T speech recognizer and determined the word accuracy of the result against both the correct answer and that given by ASR using the exact models. Figures 4–5 plot the results for two different NAB language models. The middle curves show the relative sizes of the models produced by $\tilde{\mathcal{D}}$ compared to the respective exact models produced by $\mathcal{D}$. The bottom curves show that recognition word accuracy did not degrade appreciably. ($\tilde{\mathcal{D}}$ with $\varepsilon = 0$ is equivalent to $\mathcal{D}$.) The top curves show that the output of ASR with the approximate models was nearly identical to the output with the exact models: we did not just substitute one mistake for another.

We repeated this experiment for twelve different beam widths, producing similar results each time.

5. CONCLUSION

The experimental evidence shows that approximate determinization produces models that are considerably smaller than the minimal formally equivalent models, with negligible effects on ASR performance. The 25–35% saving in model space reduces the memory consumption by the recognizer as well as the off-line model-building time. We saw no effect on ASR time, and this deserves further study. We surmise that $\tilde{\mathcal{D}}$ shrank parts of the model that were not explored during recognition. As bigger models are required, we expect the size reduction achieved by $\tilde{\mathcal{D}}$ to have a positive effect on caching and paging.

6. REFERENCES

[1] J. Berstel. *Transduction and Context-Free Languages*, volume 38 of *Leitfaden der angewandten Mathematik und Mechanik LAMM*. Springer-Verlag, 1979.

[2] J. Berstel and C. Reutenauer. *Rational Series and Their Languages*, volume 12 of *EATCS*. Springer-Verlag, 1988.

[3] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, 1974.

[4] P. Kenny, R. Hollan, V. N. Gupta, M. Lenning, P. Mermelstein, and D. O'Shaughnessy. A*-Admissible heuristics for rapid lexical access. *IEEE Trans. Speech and Audio Proc.*, 1:49–57, 1993.

[5] R. Lacouture and R. D. Mori. Lexical tree compression. In *Proc. 2nd EUROSPEECH*, volume 2, pages 581–4, 1991.

[6] M. Mohri. Finite-state transducers in language and speech processing. *Comp. Ling.*, 23(2):269–311, 1997.

[7] F. Pereira and M. Riley. Speech recognition by composition of weighted finite automata. In *Finite-State Language Processing*. MIT Press, 1997.

[8] F. Pereira, M. Riley, and R. Sproat. Weighted rational transductions and their application to human language processing. In *Proc. ARPA HLT*, pages 249–54, 1994.

[9] M. D. Riley, A. Ljolje, D. Hindle, and F. C. N. Pereira. The AT&T 60,000 word speech-to-text system. In *Proc. 4th EUROSPEECH*, volume 1, pages 207–210, 1995.