SYSTEM COMBINATION FOR IMPROVED AUTOMATIC GENERATION OF N-BEST PROPER NOUNS PRONUNCIATION

Richard Duncan
Mississippi State University
Mississippi State, Mississippi 39762, USA
e-mail: richard.duncan@ieee.org
Ph (601) 325-3149 - Fax (601) 325-3149

ABSTRACT

Proper nouns present a challenging problem for current speech recognition technology since they often do not follow typical letter-to-sound conversion rules. Several different automated methods, Boltzmann machines, Decision Trees, and Recurrent Neural Networks have been attempted recently, yet no single system has achieved an acceptable error rate. Since the project goal is the generation of pronunciation dictionaries for speech recognition, however, we can easily combine the multiple outputs of the multiple systems and use the total database coverage as our scoring metric. For generating at least one correct pronunciation for all names, combining all systems gives us a 19.6% error rate, a 23.1% absolute reduction over the best previous system. For generating every pronunciation in the database the combined system rates at 29.1%, a 23.6% reduction.

1. INTRODUCTION

Many applications for speech recognition technology require the systems to understand proper nouns. In order to recognize proper nouns a system must be in place to generate reasonable accurate pronunciation networks for these words. This is a challenging problem since many proper nouns, especially names, do not follow the letter-to-sound rules common across the rest of the vocabulary. Furthermore, multiple valid pronunciations evolve due to various socio-linguistic phenomena, so a robust recognition system needs access to all variants.

The first application for which proper noun pronunciation took center stage was directory assistance. These systems used extensive handwritten rule sets to generate accurate pronunciations of names. While they performed fairly well for the application, the systems were incapable through design of generating the multiple pronunciation variants needed for robust recognition. For regular words in a language, translation from text-only spelling to pronunciation is fairly strait forward and can be accomplished with an appropriately large rule set. Proper nouns are far more complicated and an impractically large rule set is needed for appropriate coverage. Therefore these rule-based these rule-based systems do not generalize well when presented with patterns not in the rule set.

The alternative to developing these handwritten rule sets is to apply data driven statistical methods. Recent studies in applying a Boltzmann machine feed-forward neural network, statistical decision trees, and a recurrent neural networks have met with mixed success. The Boltzmann machine feed-forward network was chosen for its ability to generate multiple pronunciations. Decision tree systems, also capable of multiple outputs, were shown to outperform the Boltzmann machine. Even though the recurrent neural network was only able to output a single pronunciation per name its first choice was nearly as accurate as the decision tree. None of these systems have performed as well as a handwritten rule set, though, so proper noun recognition applications have not noticed an appreciable change in available technology.

A hot topic in machine learning is that combining multiple classifiers can lead to significantly better results than any of the classifiers alone[2]. We can exploit that the different classifiers use different information sources and hence make different errors on the same test set. This means that for a given test utterance the chances of all systems making the wrong choice is greatly reduced. System combination is even more appropriate for pronunciation generation since it enriches the list of pronunciations and broadens the coverage of our pronunciation networks.

The speech recognition community is also adopting system combination through post-processing. The ROVER (Recognizer Output Voting Error Reduction) system combines the hypotheses of multiple recognition systems into a single word transition network through iterative dynamic programming alignments. It then produces a composite system score by finding the minimum cost path through this network. This allows ROVER to choose system A's correct hypothesis at the beginning of an utterance and system B's hypothesis at the end of the utterance.

This paper first provides an overview of these different pronunciation generation methods. It then explores the complementarity of the errors to determine if the different systems are getting the same names wrong. Finally we introduce a framework for a system that uses multiple algorithms to automatically generate the robust pronunciation networks needed for proper noun recognition.

2. PREVIOUS RESULTS

Our manually transcribed database of pronunciations contains 18,494 names and 25,648 pronunciations. The test set for this database contains 3489 names and 4579 pronunciations, mostly preserving the ratio of pronunciations to names within 5%. Since we are working towards multiple pronunciations there are three performance cases for each name:

- *all correct* — all reference pronunciations were generated
- *some correct* — at least one pronunciation was generated
- *no correct* — none of the reference pronunciations were generated.

This setup has been simplified in the new evaluations. We now consider two numbers, single pronunciation and multiple pronunciations. Single pronunciations, denoted as (1-found), is the percentage of names that fall in the *no correct* category. The coverage of multiple pronunciations is scored by comparing the total number of correct pronunciations generated to the total number of pronunciations, denoted as (N-found).

There is also some confusion in the previously published results as to what numbers are closed loop testing and what numbers are open loop testing. From a machine-learning standpoint the closed-loop performance does tell you how well the system has "remembered" the training data, but rule-based systems should do this very well already (probably better than these new automatic systems do). Thus only open loop performance can be quoted for these new systems as an indicator of technology improvement.

A. Boltzmann Machine Neural Network

The Boltzmann machine is a feed-forward neural network capable of efficiently producing multiple-outputs. This systems application to proper-noun pronunciation is discussed in [5,6]. The best published performance for open-loop testing is 66.9% error, but with the new experimental setup this number is higher.

context	1-found	N-found
3	83.5%	87.2%
5	86.8%	90.0%
7	90.0%	92.1%

Table 1: Boltzmann machine error rates

B. Decision Trees

A decision tree classifies data by partitioning the data into subsets. This approach is capable of handling nonlinear decision regions [4,8]. In order to use a decision tree classifier for pronunciation generation a tree is first trained to learn the most probably output phoneme sequence given a limited context of letters. Full pronunciations are generated by sliding a context window through a name and combining the outputs. The best number quoted is 39% error, which must be some combination of closed-loop and open-loop evaluation.

context	1-found	N-found
3	43.3%	52.8%
5	63.4%	69.5%
7	81.6%	85.0%

Table 2: Decision tree error rates

C. Recurrent Neural Network

A recurrent neural network is different from a feed-forward system in that it allows feedback from the output. Little is known about the actual system used to generate pronunciations, but the performance of the system was comparable to the best decision tree system. Le quotes a performance number of 40%, compared to her decision tree number of 39%[8].

context	1-found	N-found
3	50.3%	61.7%
5	42.8%	55.9%
7	42.7%	55.8%

Table 3: Recurrent neural network error rates

D. General Rule-based Synthesis

The final system used for comparison was the public domain rsynth package, a general purpose text to speech system [8]. A weakness of these evaluations is that our rule based comparison point is rsynth, a system not explicitly designed for proper nouns. The use of a better synthesis package such as Festival [7] should lead to a better baseline.

1-found	N-found
74.8%	80.7%

Table 4: rsynth error rates

3. EVALUATION

Besides the scoring metrics described above, further analysis was performed to find the complementarity of the errors. Following Brill's model [2], we define the complementarity of errors for two system to be:

$$Comp(A, B) = \left(1 - \frac{\text{\# of common errors}}{\text{\# of errors in A}}\right) \quad (1)$$

This measurement shows the percentage of time when algorithm A does not generate a correct pronunciation and algorithm B does generate a correct pronunciation. This measure of overlap is very important for evaluating system combination. It could be the case that some pronunciations are just more difficult than others and hence no system can find accurate pronunciations given the training data. The best two individual systems are a

	bm_3	bm_5	bm_7	dt_3	dt_5	dt_7	rnn_3	rnn_5	rnn_7	rsynth
bm_3	0.000000	0.075815	0.052487	0.526929	0.346827	0.167753	0.441852	0.535163	0.539279	0.225729
bm_5	0.110598	0.000000	0.049521	0.545724	0.350941	0.174645	0.474084	0.552988	0.555299	0.226807
bm_7	0.119541	0.082244	0.000000	0.553395	0.354797	0.173733	0.479439	0.553395	0.554989	0.232069
dt_3	0.087963	0.089947	0.073413	0.000000	0.222883	0.097884	0.161376	0.331349	0.332010	0.169312
dt_5	0.139240	0.111212	0.084991	0.468806	0.000000	0.070072	0.395569	0.407776	0.412296	0.208861
dt_7	0.147875	0.121883	0.089568	0.520899	0.277485	0.000000	0.453811	0.514225	0.510010	0.229013
rnn_3	0.073990	0.093341	0.070575	0.278315	0.239044	0.114969	0.000000	0.303358	0.306204	0.163347
rnn_5	0.092431	0.093101	0.061621	0.322840	0.122572	0.073677	0.180174	0.000000	0.024782	0.153382
rnn_7	0.098658	0.095973	0.063087	0.322147	0.127517	0.063758	0.181879	0.022819	0.000000	0.155033
rsynth	0.134918	0.102338	0.076658	0.518589	0.329245	0.158681	0.436566	0.515523	0.517439	0.000000

Table 5: Complementarity of errors between different pronunciation generation engines

decision tree with a context width of 3 letters (38%) and a recurrent neural network with a context width of 7 letters (42%). These two systems have an error complementarity of 33%, which means that a significant chunk of the error sets are disjoint.

The very high complementarity numbers for the Boltzmann machine must be taken with a grain of salt, however, since the absolute error rate for this algorithm is over 80%. In order to create a useful metric with larger base error rates, these numbers need to be conditioned by the absolute error rates of the system in question. The next section of this paper describes the actual error rates of combined systems.

4. RESULTS

In order to determine the performance of a combined system the outputs are pooled together. That is, the combined system *bm_3* and *rnn_7* is said to have *1-found* pronunciation if either system has at least one pronunciation listed as output. It should be noted that with this metric it is impossible for a combined system to have worse performance than either of its subcomponents.

The results of combining systems are shown in table 6 below.

System	1-found	N-found
rsynth	74.8%	80.7%
bm (all)	74.7%	80.0%
dt (all)	32.3%	41.6%
rnn (all)	34.4%	46.9%
bm ∪ dt	28.0%	37.3%
bm ∪ rnn	29.6%	41.9%
bm ∪ rsynth	59.4%	67.5%
dt ∪ rnn	24.2%	34.3%
all	19.66%	29.2%
all ∩ bm	21.4%	31.3%
all ∩ dt	26.3%	38.2%
all ∩ rnn	25.0%	34.1%
all ∩ rsynth	21.5%	31.4%

Table 6: Combined system error rates

Combining all systems yields a drastic improvement in overall system performance. Only 19.7% of the input names lack a single correct pronunciation, a 23% reduction in absolute error.

The Boltzmann machine and rsynth do not considerably affect performance, but both systems do seem to get a few words correct that both the recurrent neural networks and the decision trees miss. Combining the Boltzmann machine and decision tree does bring down the overall performance about 4.3% from 32.3% absolute, something which could be predicted from the complementarity table row 4 position 1. This is not absolutely clear since the complementarity table does not have entries for partial combined entries.

5. DISCUSSION

The ease with which system performance seems to increase begs for an obvious retort — how can simply creating larger lists be a valid metric? The goal of this project is to build pronunciation networks. If there is not a valid path in the network for a given pronunciation the recognizer will most likely misrecognized a name. Invalid paths, however, can be pruned away through acoustic scores. While smaller networks will improve system performance somewhat a missing pronunciation is far more undesirable. Recognition results will in fact be the only way to prove the validity of this work.

It is evident in the data that the different systems are most definitely making different mistakes. This implies that they are using different pieces of information to generate the pronunciations. One area of study is to determine why the individual systems are not using this information.

A full least-cost network is not yet in use. By combining the different pronunciations into a single phone-transition-network by aligning different contexts more paths can be generated. This type of network building should allow more pronunciations to be valid paths and further reduce the error rate. Furthermore, the likelihood scores produced by some of the systems are completely ignored. These scores could be added as weights to the phone-transition-networks to further aid the recognition engine.

6. REFERENCES

[1] Fiscus, Jonathan G. 1997. A Post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding held on December 1997*. 347-54.

[2] Brill, Eric, and Jun Wu. 1998. Classifier combination for improved lexical disambiguation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics at the Universite de Montreal, Montreal, Quebec, Canada on August of 1998*. ACL / Morgan Kaufmann Publishers. Volume 1, 191-5.

[3] Deshmukh, Neeraj. 1999. Maximum likelihood estimation of multiple pronunciations for proper nouns. Ph.D. Dissertation Proposal, *Institute for Signal and Information Processing, Mississippi State University*, Mississippi State, Mississippi.

[4] Ngan, Julie. Aravind Ganapathiraju and Joe Picone. 1998. Improved surname pronunciations using decision trees. In *Proceedings of the ICSLP held in Sydney, Australia on November 1998*, 3285-8.

[5] Deshmukh, Neeraj, Audrey Le, Julie Ngan, Jonathan Hamaker and Joe Picone. 1997. An advanced system to generate multiple pronunciations of proper nouns. In *Proceedings of the IEEE ICASSP held in Munich, Germany on April 1997*. 1467-70.

[6] Deshmukh, Neeraj, Mary Weber, and Joe Picone. 1996. Automated generation of N-best pronunciations of proper nouns. In *Proceedings of the IEEE ICASSP held in Atlanta, Georgia on May 1996*, 283-6.

[7] The festival speech synthesis system. 2000. http://www.cstr.ed.ac.uk/projects/festival/. *The Centre for Speech Technology Research, University of Edinburgh*, Scotland (Accessed 19 April 2000).

[8] Le, Audrey. 1998. Bayesian Decision Tree for Classification of Nonlinear Signal Processing Problems," http://www.isip.msstate.edu/publications/seminars/masters_oral/1998/decision_tree_bayes/index.html. *Master of Science Special Project Presentation, Mississippi State University* (Accessed 20 April 2000).