IMPROVED SURNAME PRONUNCIATIONS USING DECISION TREES

Julie Ngan, Aravind Ganapathiraju, Joseph Picone

Institute for Signal and Information Processing
Department of Electrical and Computer Engineering
Mississippi State University, Mississippi State, Mississippi 39762
{ngan, ganapath, picone}@isip.msstate.edu

ABSTRACT

Proper noun pronunciation generation is a particularly challenging problem in speech recognition since a large percentage of proper nouns often defy typical letter-to-sound conversion rules. In this paper, we present decision tree methods which outperform neural network techniques. Using the decision tree method, we have achieved an overall error rate of 45.5%, which is a 35% reduction over the previous techniques. Our best system is a binary decision tree that uses a context length of 3 and employs information gain ratio as the splitting rule.

1. INTRODUCTION

Proper noun recognition is a critical component in achieving high performance speech recognition. Further, there is renewed interest in this problem with the recent decision by the LVCSR community to adopt the named entity task as the next step towards a speech understanding framework for common evaluations. In order to recognize proper nouns, an ability to generate accurate pronunciation networks is required. This problem is particularly challenging because a large percentage of proper nouns, such as surnames, have no obvious letter-to-sound mapping rules that can be used to generate the pronunciations. Moreover, many proper nouns have multiple valid pronunciations that evolve as a product of various socio-linguistic phenomena, and the system needs to generate accurate pronunciation networks to cover all the accepted pronunciation variants for correct identification. Classical rule-based systems are inherently unsuitable for this task as they generate only a single pronunciation.

Previous attempts based on stochastic neural networks to generate pronunciations from letter context [1, 2] have met with mixed success. In this paper, we will present an improvement in the state-of-the-art on this task using decision tree technology. Statistical decision trees (DT) have recently emerged as a versatile and data-driven classification tool for complex, non-linearly separable data. Based on the response to a series of simple multi-valued questions, decision trees can efficiently and accurately generate classification clusters of highly complex decision boundaries. They also provide insights into the underlying phenomena and facilitate accurate prediction of events that pose problems for analytic clustering methods. For instance, phonetic decision trees successfully employ phonological knowledge that cannot be otherwise incorporated to perform efficient state-tying of Hidden Markov Models (HMMs) for speech recognition [3].

2. DECISION TREES

Decision trees are generated in a top-down fashion using the statistics of the training data. At each node, the tree iteratively splits the distribution of the training data to maximize its likelihood by evaluating each question. Therefore, decision trees require a large amount of training data to model a distribution that is representative of the problem space. However, public domain decision tree software packages such as IND [4] and ID3 [5] are limited in the number of classes, attributes as well as the nature and range of attribute values. These, as well as the bounds on the amount of data they can process, make them impractical for large scale problems such as generation of proper noun pronunciations.

In order to overcome such problems with existing software, we are developing a public domain decision tree software package as part of our speech recognition toolkit. Written entirely in object-oriented C++, it is tailored to handle large amounts of training data and is equipped with the ability to support an unlimited number of attributes, attribute values, and classes. It is also designed to handle a user-defined combination of splitting, stopping, pruning, and smoothing algorithms. Furthermore, our software allows data tagging, which enables each attribute to be selected or deselected from the attribute file without having to reformat the training data for each experiment.

For pronunciation generation, the decision tree system is trained using a set of name-pronunciation pairs. Using a sliding window of a fixed context length, n-tuple of letters of the proper noun spelling are created with a corresponding phoneme from the pronunciation associated with it. Each sequence can thus be treated as an individual training sample. For example, using a context length of 5, the name *Matt* (with a pronunciation of *m@t*) will generate training sequences as illustrated in Figure 1. The system thus learns the statistical relationship between each n-tuple of letters and its corresponding phoneme.

For recognition, the system converts the input names into n-tuples of an equal context size. Using the probabilistic model formed by the splitting algorithm during training, it generates the most likely phoneme for each n-tuple input. Figure 2 shows a simplified snapshot of a decision tree model (with context length 5) used in our system. At each node, a yes/no question regarding the context is asked and the corresponding path is taken until a terminal node is reached. Encoded at each terminal node are the output classes and their statistics, which form a list of probable phonemes. The phoneme strings generated in this fashion are then reformatted to create the pronunciation of the full name.

3. PROPER NOUNS DATABASE

One aspect of proper noun pronunciation generation that makes it particularly challenging and timely is that there are no existing proper noun databases that include extensive lists of plausible alternate pronunciations for a demonstrative sample of proper nouns. In order to train and evaluate the system, we have compiled an extensive hand-transcribed phonetic proper noun databaseand placed it in the public domain [6]. This pronunciation dictionary consists of approximately 18,500 surnames and close to 24,000 name-pronunciation pairs. Further, this database adheres to the Worldbet [7] pronunciation alphabet and represents a reasonably diverse set of names from a wide variety of ethnic origins.

Since the decision tree model is designed to generate a phoneme symbol for each context window, it is important to produce accurate letter-to-phone alignments for the entire word. The database uses a dynamic programming algorithm to perform this alignment automatically. For letters that have no corresponding phoneme in the pronunciation, it sets a blank phoneme "_" [8]. For example, *Wright* is transcribed and aligned as '*_ 9r aI _ _ t*'. After the phoneme alignment, the training and evaluation data sets are generated using a fixed-length context.

_	**_**	**m**	**a**	**t**	**m**
_	**m**	**a**	**t**	**t**	**@**
m	**a**	**t**	**t**	**_**	**t**
a	**t**	**t**	**_**	**_**	**_**

Figure 1. Context alignment for the name Matt.

4. EXPERIMENTS AND RESULTS

We have devised three categories to measure the mapping between the reference and hypothesis pronunciations. *All correct* represents that all the reference pronunciations for the proper noun are covered by the hypothesis pronunciations generated by the system; *some correct* represents that only some of the reference pronunciations are covered by the hypothesis pronunciations; and *no correct* represents that none of the reference pronunciations

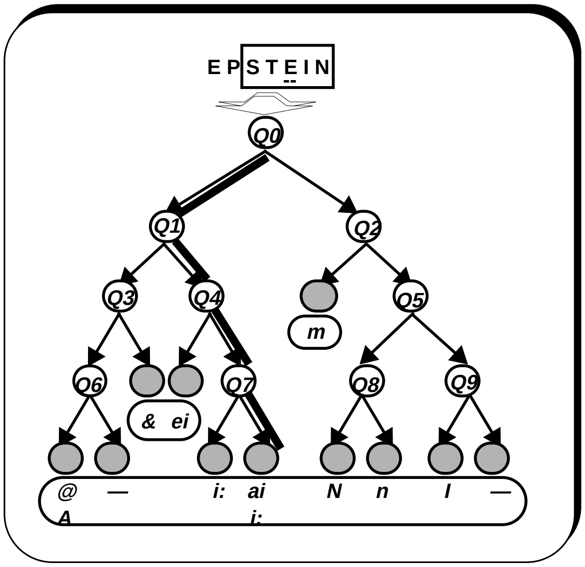

Figure 2. A typical statistical decision tree for automatic generation of pronunciations of proper nouns.

match the hypothesis pronunciations. The *no correct* category indicates the name error rate of the system.

4.1. Pilot Experiments

We used a data set consisting of 128 four-letter names to perform closed-loop tests to gauge the training and evaluation paradigms of the system. After evaluating several algorithms such as two-ing [9], Bayesian splitting and smoothing [10], information gain [11], and gain ratio [12], we found the best overall system to be a binary, univariate tree that is split using the maximum gain ratio and the average information gain per split. The performance of this decision tree system in comparison with the baseline ANN system is shown in Table 1.

4.2. Four-letter Names

To study the impact of the scale of the problem on the decision tree system, the next set of evaluations was conducted on a subset of the dictionary that comprised of all the four-letter surnames. This formed a training set of 1,617 surnames and a test set of 408 names. The results for evaluations on this data are in Table 2. It is evident that the DT approach yields a lower misclassification rate and shows substantial improvement over the neural networks.

4.3. Full Evaluations

The performance of the decision tree system was next evaluated on the full proper noun dictionary. The complete database was partitioned into three overlapping training sets of approximately 19,500 name-pronunciation pairs and corresponding three held-out test sets of approximately 4,500 names, thus creating a cross-validation paradigm to ensure accurate results of the decision tree system.

A context of length three was used to train the system. The misclassification rate of the decision tree compared with that of the Boltzmann machine is summarized in Table 3; A more detailed summary of the DT results is shown in Table 4.

Comparing the results from the Boltzmann machine and our decision tree system, it can be seen that the decision trees method has achieved an overall error rate reduction of 35%, thus proving that the decision trees system is more robust for automatically generating pronunciations of proper nouns.

System	Phoneme error rate	Name error rate
Boltzmann machine	11.13%	35.94%
Decision tree	4.10%	14.06%

Table 1: Misclassification rate for the closed-loop 128 four-letter names using neural networks and decision trees.

System	Phoneme error rate	Name error rate
Boltzmann machine	20.76%	52.13%
Decision tree	17.52%	44.85%

Table 2: Misclassification rate for the open-loop 1617 four-letter names using neural networks and decision trees.

System	Phoneme error rate	Name error rate
Boltzmann machine	37.88%	70.44%
Decision tree	13.28%	45.50%

Table 3: Summarized misclassification rate for the full proper noun data set using neural networks and decision trees.

data set	all correct	some correct	no correct
1	30.43%	23.42%	46.15%
2	30.89%	23.56%	45.55%
3	30.04%	24.46%	45.50%

Table 4: Detailed decision tree performance on the three complete proper noun database partitions.

From the results on the three partitions, note that the best decision tree configuration yields an error rate of 45.5%. It should also be observed that the results are consistent over the three partitions, which indicates that the decision tree method does not memorize the training data but generalizes well.

We have yet to evaluate the system to produce N-best proper noun pronunciations. Our current decision tree network generates a single pronunciation per proper noun. However, these results are comparable to the results achieved in our previous DT based work with multiple output pronunciations. In [8], we report an error rate of 47.13% and 42.53% using 5-best and 10-best pronunciations respectively. We project a further decrease of error rate for our system generating multiple pronunciations.

5. CONCLUSIONS AND FUTURE WORK

We have shown that using decision trees for data classification and clustering is promising for proper noun pronunciation generation. This techniques has the potential to generate more accurate multiple pronunciations than previously attempted methods. Using decision trees, we have achieved an error rate reduction of 35% over neural network systems.

However, the accuracy of a decision tree depends highly on the training data. The highly nonlinear and conflicting nature of the pronunciations will require a larger training database with more complete coverage of the pronunciation combinations. Our future work will involve expanding the dictionary in this fashion, as well as incorporating a back-off algorithm such to allow flexible context lengths to generate more accurate pronunciations. Moreover, other less common decision tree splitting and pruning algorithms will also be implemented into our system. Addition of pruning algorithms will ensure good generalization ability of the system.

This is the first public domain decision tree package that has been successfully applied to such a large speech-related classification task on which other nonlinear classifiers have failed. We envision our decision tree software package will be a useful tool in future data classification research for the speech recognition community. The pronunciation dictionary, as well as the decision tree and neural network software developed for pronunciation generation has been placed in the public domain [6].

6. REFERENCES

1. N. Deshmukh and J. Picone, "Automatic Generation of N-Best Pronunciations of Proper Nouns," submitted to the *IEEE Transactions on Speech and Audio Processing*, November 1996.

2. N. Deshmukh, M. Weber, and J. Picone, "Automated Generation of N-Best Pronunciations of Proper Nouns," *Proceedings of ICASSP '96*, pp. I283-I286, Atlanta, GA, May 1996.

3. J. J. Odell, "*The Use of Decision Trees with Context Sensitive Phoneme Modelling*," MPhil Thesis, Cambridge University Engineering Department, 1992.

4. W. Buntine and R. Caruana. "Introduction to IND Version 2.1 and Recursive Partitioning," software manual, NASA Ames Research Center, Moffet Field, CA, 1992.

5. W. Buntine, "*Tree Classification Software*," The Third National Technology Transfer Conference and Exposition, Baltimore, MD, December 1992.

6. N. Deshmukh et al, "Automatic Proper Noun Pronunciations," *http://WWW.ISIP.MsState.Edu/resources/technology/projects/1997/nbest_pronunciations*, Institute for Signal and Information Processing, Mississippi State University, 1997.

7. J. L. Hieronymus. "ASCII Phonetic Symbols for the World's Languages: Worldbet," *Technical Memo,* AT&T Bell Laboratories, 1994.

8. N. Deshmukh, J. Ngan, J. Hamaker, and J. Picone, "An Advanced System to Generate Multiple Pronunciations of Proper Nouns," *Proceedings of ICASSP '97*, vol. 2, pp. 1467-1470, Munich, Germany, April 1997.

9. L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.

10. W. Buntine, *A Theory of Learning Classification Rules*, Ph.D. thesis, University of Technology, Sydney, 1991.

11. I. K. Sethi and G. Sarvarayudu, "*Hierarchical Classifier Design Using Mutual Information,*" IEEE Trans Patt. Anal. Mach. Intell., vol. PAMI-4, pp. 441-445, 1982.

12. J. R. Quinlan, *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.