IMPROVED NAME RECOGNITION WITH META-DATA DEPENDENT NAME NETWORKS

Sameer R. Maskey[†]*, Michiel Bacchiani*[‡]*, Brian Roark*[‡]*, and Richard Sproat*[◦]

[†]Dept. of Computer Science, Columbia University, 1214 Amsterdam Avenue New York, NY 10027
smaskey@cs.columbia.edu
[‡]AT&T Labs-Research, 180 Park Ave., Florham Park, NJ 07932, USA
{michiel,roark}@research.att.com
[◦]Departments of Linguistics and ECE, University of Illinois at Urbana-Champaign
Foreign Languages Building 4103, 707 South Mathews Avenue, MC-168 Urbana, IL, 61801
rws@uiuc.edu

ABSTRACT

A transcription system that requires accurate general name transcription is faced with the problem of covering the large number of names it may encounter. Without any prior knowledge, this requires a large increase in the size and complexity of the system due to the expansion of the lexicon. Furthermore, this increase will adversely affect the system performance due to the increased confusability. Here we propose a method that uses meta-data, available at runtime to ensure better name coverage without significantly increasing the system complexity. We tested this approach on a voicemail transcription task and assumed meta-data to be available in the form of a caller ID string (as it would show up on a caller ID enabled phone) and the name of the mailbox owner. Networks representing possible spoken realization of those names are generated at runtime and included in network of the decoder. The decoder network is built at training time using a class-dependent language model, with caller and mailbox name instances modeled as class tokens. The class tokens are replaced at test time with the name networks built from the meta-data. The proposed algorithm showed a reduction in the error rate of name tokens of 22.1%.

1. INTRODUCTION

Many applications, such as directory assistance and name dialers, require an Automatic Speech Recognition (ASR) system capable of accurate name transcription. Building such a system is complex due to the very large number of different names it may encounter. An additional complicating factor is the pronunciation of names which can vary significantly among speakers. As a result, ASR research on name recognition has received a fair amount of attention, e.g. [1, 2, 3, 4, 5]. In [1], the feasibility of a directory assistance application with as many as 1.5 million names was investigated. It showed that recognition accuracy drops approximately logarithmically with increasing vocabulary size. A significant degradation in performance with increasing lexicon size was also noted in [4], however it showed that a larger lexicon, allowing more diverse pronunciations, was beneficial. Most efforts have focused on soliciting more detailed speech input from the user in the form of spelling, and have shown that this improves the system performance. In [2], neural networks are used to focus the search on the most discriminative segments in a multi-pass approach. Along similar lines, it is shown in [5] that accuracy can be improved by incorporating confidence scores into the decision process.

Common among all previous work is that the coverage issue was addressed by increasing the vocabulary size. The increased confusability introduced by that increase is then addressed by more complex search and acoustic modeling. Something that is not taken into account in such a modeling approach is the prior probability distribution across names. Indeed, if no additional information is available, a uniform (or context independent frequency weighted) distribution across names is a reasonable estimate. However, in most contexts, a very small subset of the possible names will account for most of the true probability mass. In other words, the distribution of names seen in the speech of a particular speaker is very unlikely to be distributed uniformly across the large list of possible names. If the subset of names that are most likely to occur in a given context are known, the system accuracy can be increased with a decrease in complexity.

In the work presented here, we focus on name recognition in a voicemail transcription task and assume context information or *meta-data* is available in the form of the name of the mailbox owner and the caller ID string from the incoming call leaving the voicemail message. Given the prevalence of caller identification provided by the phone

companies, this appears to be a reasonable assumption. First in section 2 we describe the voicemail database. Then in section 3 we describe how the meta-data is used to condition the ASR system. Section 4 describes the experimental results obtained using this approach and section 5 discusses these results.

2. DATABASE

The transcription experiments were conducted on a 100 hour corpus of voicemail messages collected from the voicemail boxes of 140 employees at AT&T. This corpus, named Scanmail, contains approximately 10,000 messages from approximately 2500 speakers. The corpus is approximately gender balanced and approximately 12% of the messages are from non-native speakers (as assessed by the labeler from listening to the speech). The mean duration of the messages is 36.4 seconds, the median is 30.0 seconds.

The messages were manually transcribed and those parts of the transcripts that identify the caller and mailbox owner were bracketed. The identifications usually occur in the beginning of the message as in

```
hi [Greeting: mister jones] this is
[CallerID: john smith] calling...
```

A 2 hour test set was chosen by randomly selecting 238 messages from the corpus. The remaining speech was used as the training set to build the acoustic and language models. In this test set, there were 317 word tokens corresponding to caller names and 219 word tokens corresponding to mailbox owner names.

3. APPROACH

Our approach to including the name meta-data into the ASR system uses a class-based language model, built at training time. This language model represents name occurrences by class tokens. Then at test time, the name meta-data is used to produce a name network that gives possible, probability weighted spoken realizations of the meta-data defined names. That name network is then included in the recognition network by a network replacement step. Section 3.1 describes the language model, section 3.2 describes the name networks and section 3.3 describes the network replacement.

3.1. Class-Based Language Model

We follow an approach similar to that in [6] for constructing class-based language models. Sequences of tokens in the training corpus that were annotated as the mailbox name or the caller name were replaced with the class labels $\langle mname \rangle$ and $\langle cname \rangle$, respectively. From this corpus, with class labels treated as words, a standard Katz backoff trigram model was built, and encoded as a weighted finite-state automaton. To make the model usable, transitions labeled with class labels must then be replaced by the sequences of words that are members of that class.

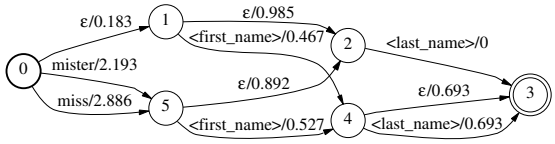

Fig. 1. Name sequence network

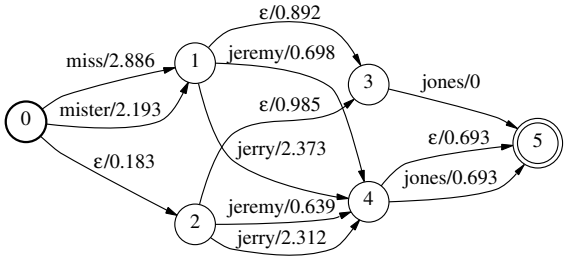

Fig. 2. Name network for Jeremy Jones

3.2. Name Network

For each voicemail message in the test set, the name of the mailbox owner was provided, and the name of the caller, if it was available, which it was for 71 percent of the test messages. For each provided name, e.g. Jeremy Jones, there are a variety of ways in which the name could be realized, e.g. Jerry Jones, Mister Jones, Jeremy, etc. This variation is the result of two random processes: first, the sequence of title, first name and last name can vary; next there can be many possible forms of the first name. From the training corpus, we estimated, for each name class, the probability of different realizations of the sequence of title, first name (regardless of form) and last name. Figure 1 shows a weighted acceptor with first name and last name labels, which represents a distribution over possible name sequences, weighted by negative log probabilities.

For the probabilities of forms of first names, we made use of an internal AT&T directory listing, which includes the full name and optional nicknames for 40,000 employees. For a given first name, we counted each nickname for people with that name, and used the maximum likelihood estimate based on these counts for the nickname given the

name[1]. For a particular caller ID, the <first_name> and <last_name> tokens in the graph in figure 1 must be replaced by the actual last name and a distribution over possible first name forms – i.e. nicknames or the full form – for the specific caller. Figure 2 shows such a weighted name sequence acceptor when the caller name is Jeremy Jones.

All occurrences of the ⟨*cname*⟩ token in the language model must then be replaced by this network, with their weights combined. This can be done with composition of finite-state transducers, as discussed in [6].

3.3. Replacement in the ASR network

The Scanmail voicemail system [7] at AT&T uses an optimized recognition network, which combines the pronunciation lexicon $\mathcal{L}$ and the grammar $\mathcal{G}$ into a single optimized finite-state transducer through off-line composition, determinization and minimization [8]. This network composition and optimization can be quite costly in space and time, and is generally done once and the result treated as a static model.

In the current scenario, this resource cannot be static, since each message can have a different mailbox and caller ID. Composing and optimizing the entire network for each message is impractical. To avoid this, we provide each name class label with a special phone symbol in the lexicon, which allows us to produce an optimized $\mathcal{L} \circ \mathcal{G}$ for the class-based $\mathcal{G}$. For each message, we produce $\mathcal{L} \circ \mathcal{G}'$ by composing the name network $\mathcal{G}'$ with the lexicon and optimizing. Every transition in the original class-based $\mathcal{L} \circ \mathcal{G}$ with a name class label (i.e. ⟨*mname*⟩ or ⟨*cname*⟩) as the output label (and hence the special phone symbol as the input label) is then replaced with the $\mathcal{L} \circ \mathcal{G}'$ for that name class, and the weights are combined appropriately. The overhead of producing the very small $\mathcal{L} \circ \mathcal{G}'$ and replacement in the large $\mathcal{L} \circ \mathcal{G}$ is relatively low.

4. EXPERIMENTAL RESULTS

We evaluated our algorithm on the 238 message Scanmail test set. This test set was drawn from the Scanmail corpus by random selection of messages. This means that for most test messages, there will be messages in the training set that were received in the same mailbox. The number of training messages received at a particular mailbox varied from 1 to 11 with an average of 3 messages per mailbox. The overlap in mailbox recipients results in an experimental setup that is likely to provide a lower error rate, especially on names, than a scenario where the test data is from mailboxes never seen in the training data. To normalize for this effect, we experimented using a different language model for each test message. The language models were constructed by excluding training messages from the same mailbox as the test message.

For the 238 test messages, the ⟨*mname*⟩ meta-data value was known for all messages but the ⟨*cname*⟩ meta-data was available for only 169 messages. For the messages that did not have the ⟨*cname*⟩ meta-data available, we used the a system that only used the ⟨*mname*⟩ class.

To evaluate the performance of the algorithm, in addition to Word Error Rates (WER) we measured the error rate on the name tokens corresponding to the ⟨*mname*⟩ and ⟨*cname*⟩ class tokens. Using the alignments produced in computing the WER, the Name Error Rate (NER) is computed as the percentage of name tokens that were labeled as an error (either a deletion or a substitution) in that alignment.

The baseline system using no name replacements had a WER of 26.6% (7233 tokens). Using the proposed algorithm replacing only ⟨*mname*⟩ tokens, the WER dropped to 26.3% (7147 tokens). When replacing both ⟨*mname*⟩ and ⟨*cname*⟩ tokens, the WER rate dropped to 26.0% (7066 tokens).

System	Word Error Rate	Name Error Rate
Baseline	26.6 %	56.9 %
⟨*mname*⟩	26.3 %	45.7 %
⟨*mname*⟩ + ⟨*cname*⟩	26.0 %	34.8 %

Table 1. *WER and NER*

The performance of the algorithm is summarized in table 1). Among the 219 name tokens corresponding to ⟨*mname*⟩ class tokens, there were 128 errors in the baseline transcripts. Using the system that did ⟨*mname*⟩ replacements, this dropped to 68 errors. Among the 317 ⟨*cname*⟩ tokens, 177 were misrecognized in the baseline recognizer output. Using the ⟨*mname*⟩ and ⟨*cname*⟩ replacement system this error rate dropped to 119 errors. The total number of misrecognized name tokens in the baseline was 305 corresponding to a 56.9% NER. Using the ⟨*mname*⟩ and ⟨*cname*⟩ replacement system, the name token error rate dropped to 187 or 34.8% NER. This is an absolute NER reduction of 22.1%.

The word error rate improvement of the of the ⟨*mname*⟩ replacement system in terms of the number of tokens was 86 which is higher than the number of corrections among ⟨*mname*⟩ tokens (60) showing that the replacement had a small beneficial effect on the words surrounding the name tokens. Similarly, for the ⟨*mname*⟩ and ⟨*cname*⟩ replacement system, the number of corrected tokens in the WER computation exceeds the number of corrected ⟨*mname*⟩ and ⟨*cname*⟩ tokens by 49 showing the same small beneficial ef-

[1]If no nickname was listed, it was counted as though the full form of the name was the nickname. In order to always allow for the full form of the name, if every observation with the name has a nickname, the full form can be given one count.

fect.

Out of the 536 name tokens corresponding to the ⟨*mname*⟩ and ⟨*cname*⟩ class tokens, 35 were out of vocabulary (OOV) word tokens. The ⟨*mname*⟩ and ⟨*cname*⟩ replacement system correctly recognized 24 (69%) of those.

We computed the runtime overhead on a 30 message, randomly selected from the test set. The average real time factor processing the messages with the baseline system was 3.8. The runtime of the ⟨*mname*⟩ replacement experiment increased this factor to 4.3 (a 13% increase). For the ⟨*mname*⟩ and ⟨*cname*⟩ replacement experiment, the average real-time factor was 4.6, a 20% increase compared to the baseline.

5. CONCLUSION

Although the decrease in overall WER was not large, names are of particular importance, so that the large reduction in name error rate is critical to both the perception and use of the system. Scanmail users have expressed a strong desire for the system to recognize these tokens correctly.

The results show that the proposed algorithm is not only useful for addressing errors that arise from OOV tokens but also improves on in-vocabulary name recognition. Where in a static system, the distribution across names may be fairly flat, the meta-data dependent system effectively provides a relatively peaked distribution for those names that correspond to allowed realizations of the given names.

Unlike previous efforts, the use of meta-data allows for the design of a system with good name coverage without a significant increase in system complexity. Although, unlike other systems, the use of meta-data incurs a run-time overhead at test time, this overhead is possibly smaller than the additional overhead incurred by a significant increase in complexity.

In contrast to systems with a static name inventory, the proposed algorithm avoids the need for manual system design when it is moved to new environment. Where a static system will likely incur an increase in the OOV rate, the proposed algorithm automatically adapts due to the run-time network generation.

6. REFERENCES

[1] C. Kamm, K.-M. Yang, C. Shamieh, and S. Singhal, "Speech recognition issues for directory assistance applications," in *Proceedings of the IEEE Workshop on Interactive Voice Technology for Telecommunications Applications*, 1994.

[2] J.-C. Junqua, S. Valente, D. Fohr, and J.-F. Mari, "An n-best strategy, dynamic grammars and selectively trained neural networks for real-time recognition of continuously spelled names over the telephone," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1995, pp. 852–855.

[3] E. McDermott, A. Biem, S. Tenpaku, and S. Katagiri, "Discriminative training for large vocabulary telephone-based name recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2000, pp. 3739–3742.

[4] Y. Gao, B. Ramabhadran, J. Chen, H. Erdogan, and M. Picheny, "Innovative approaches for large vocabulary name recognition," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 53–56.

[5] Y.-F. Liao and G. Rose, "Recognition of chinese names in continuous speech for directory assistance applications," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002, pp. 741–744.

[6] Cyril Allauzen, Mehryar Mohri, and Brian Roark, "Generalized algorithms for constructing language models," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 40–47.

[7] Michiel Bacchiani, "Automatic transcription of voicemail at AT&T," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[8] Mehryar Mohri and Michael Riley, "Weighted determinization and minimization for large vocabulary speech recognition," in *Proceedings of Eurospeech*, 1997.