Automatic Acquisition of Names Using Speak and Spell Mode in Spoken Dialogue Systems

Grace Chung†, Stephanie Seneff‡ and Chao Wang‡*

†Corporation for National Research Initiatives
1895 Preston White Drive, Suite 100, Reston, VA 22209
gchung@cnri.reston.va.us

‡Spoken Language Systems Group
MIT Laboratory for Computer Science
200 Technology Square, Cambridge, MA 02139
{seneff,wangc}@sls.lcs.mit.edu

Abstract

This paper describes a novel multi-stage recognition procedure for deducing the spelling and pronunciation of an open set of names. The overall goal is the automatic acquisition of unknown words in a human computer conversational system. The names are spoken and spelled in a single utterance, achieving a concise and natural dialogue flow. The first recognition pass extracts letter hypotheses from the spelled part of the waveform and maps them to phonemic hypotheses via a hierarchical sublexical model capable of generating grapheme-phoneme mappings. A second recognition pass determines the name by combining information from the spoken and spelled part of the waveform, augmented with language model constraints. The procedure is integrated into a spoken dialogue system where users are asked to enroll their names for the first time. The acquisition process is implemented in multiple parallel threads for real-time operation. Subsequent to inducing the spelling and pronunciation of a new name, a series of operations automatically updates the recognition and natural language systems to immediately accommodate the new word. Experiments show promising results for letter and phoneme accuracies on a preliminary dataset.

*The research at CNRI was supported by DARPA under contract number N66001-00-2-8922, monitored through SPAWAR Systems Center, San Diego. The research at MIT was supported by DARPA under contract number NBCH1020002 monitored through the Dept. of the Interior, National Business Center, Acquisition Services Div., Fort Huachuca, AZ.

1 Introduction

Spoken dialogue systems are emerging as an effective means for humans to access information spaces through natural spoken interaction with computers. These systems are usually implemented in such a way that their knowledge space is static, or is only augmented through human intervention from the system developers. A significant enhancement to the usability of such systems would be the ability to automatically acquire new knowledge through spoken interaction with its end users. Such knowledge would include both the spelling and pronunciation of a new word, as well as an understanding of its usage in the language (e.g., a semantic category). However, this is a difficult task to carry out effectively, challenging both with regard to the automatic acquisition of the sound-to-letter mapping from typically telephone-quality speech, and the system level aspect of integrating the usually off-line activities of system upgrade while seamlessly continuing the conversation with the user.

The research reported here is concerned with the acquisition of the user's name, which is entered via a "speak and spell" mode, spoken sequentially for the first and last names respectively. It is our belief that this would be the most natural way for the user to enter the information, and therefore research has been focused on designing a framework to support that model. Acquiring names is particularly difficult, not only because English is known to have highly irregular letter-to-sound rules, but also because American names come from a diverse collection of language groups. With the speak and spell entry mode, there are additional issues of locating the boundary between the spoken and spelled portions of the utterance, and of formulating a joint solution.

The framework for acquiring new names is applied to an enrollment phase of an existing spoken dialogue system, the ORION task delegation system (Seneff et al., 2000). ORION allows users to specify tasks to be completed off-line, and to later be delivered to the user at a designated time, via either telephone or e-mail. To en-

ter the enrollment phase, the user calls the ORION system and responds to the prompt by saying, "new user," which causes the system to enter a system-initiated subdialogue soliciting a number of facts to be entered into a form that will represent the system's future knowledge of this individual. The system solicits the user's full name, a password for security measures, their work, home, and cell phone numbers, and their e-mail address, finally asking for the current time in order to establish the user's reference time zone.

After all of the information has been entered, the system confirms its proposed spellings for the names, and, if verified by the user, automatically launches a system update that enters this new information into both the speech recognition component and the natural language (NL) grammar. Thus, the next time the user calls the system, they will be able to log on by speaking their name and password. If the user rejects the proposed spelling, the system solicits further input from them, in the form of a telephone keypad entry of the spelling of the misrecognized name. This information is then incorporated into the search to propose a final hypothesis.

Central to our methodology is the application of ANGIE (Seneff et al., 1996), a hierarchical framework capturing subword structure information, employed here to predict phoneme-grapheme mappings. In ANGIE, corpus-based statistical methods are combined with explicit linguistic information to generalize from the observation space to unseen words.

Our approach extends work reported earlier (Chung and Seneff, 2002), in which spelling and pronunciation of unknown names are extracted from spoken input with the additional constraint of telephone keypad input. This work distinguishes itself in that, instead of requiring telephone keypad entries, a user is asked to speak and spell their name within a single utterance. The novelty lies in the use of a multi-stage recognizer, where the first stage proposes a letter graph derived from the spelled portion of the waveform. A second recognition pass searches the pronounced name part of the waveform; this final search is constrained by a phoneme space derived from the letter graph via ANGIE letter-to-sound mappings.

In the following, previous related work is outlined in Section 2. Sections 3 details the technology of sound-to-letter acquisition, and the techniques used to implement a recognition engine to serve our unique needs. Section 4 is primarily concerned with the engineering aspects for the real time implementation. Section 5 describes some evaluation results. This paper concludes with a summary and a look to the future in Section 6.

2 Previous Work

In the past, many researchers have worked on letter-to-sound algorithms for text-to-speech conversion (Damper et al., 1998). More recently, research is beginning to emerge in bi-directional sound-letter generation and phoneme-to-grapheme conversion. These topics are important for application to speech recognition, for the purpose of automatically transcribing out-of-vocabulary (OOV) words at the spoken input. In (Meng et al., 1996), a hierarchical approach was used for bi-directional sound-letter generation. On the Brown Corpus, it achieves word accuracies of 65% for spelling-to-pronunciation and 51% for pronunciation-to-spelling. Rentzepopoulos (Rentzepopoulos and Kokkinakis, 1996) describes a hidden Markov model approach for phoneme-to-grapheme conversion, in seven European languages on a number of corpora. The algorithm gave high accuracies when applied to correctly transcribed words but was not applied to real recognition output. The work of Marchand and Damper (Marchand and Damper, 2000) addresses both phoneme-to-grapheme and grapheme-to-phoneme conversion using a fusion of data-driven and pronunciation-by-analogy methods, obtaining word accuracies of 57.7% and 69.1% for phoneme-to-grapheme and grapheme-to-phoneme experiments respectively. These were performed on a corpus of words from a general dictionary.

Some work has focused on proper names, since names are a particularly challenging open set. In (Ngan et al., 1998), the problem of generating pronunciations for proper names is addressed. A 45.5% word error rate is reported on a set of around 4500 names using a decision tree method. Font Llitjos (Font Llitjos and Black, 2001) reports improvements on letter-to-sound performance on names by adding language origin features, reporting 61.72% word accuracy on 56000 names. Galescu (Galescu and Allen, 2002) addresses bi-directional sound-letter generation using a data-driven joint n-gram method on proper nouns, yielding around 41% word accuracy for letter-to-sound and 68% word accuracy for sound-to-letter.

Few have attempted to convert a spoken waveform with an unknown word to a grapheme sequence. Using a Dutch corpus, Decadt et al. (Decadt et al., 2002) use a memory-based phoneme-to-grapheme converter to derive graphemic output from phonemic recognition hypotheses. Results showed 46.3% accuracy on training data but only 7.9% accuracy on OOV recognition test data. In a German system, Schillo (Schillo et al., 2000) built a grapheme recognizer for isolated words, towards the goal of unconstrained recognition in German. Accuracies attained are up to 72.89% for city names.

3 Approach

The approach adopted in this work utilizes a multi-pass strategy consisting of two recognition passes on the spoken waveform. The first-stage recognizer extracts the

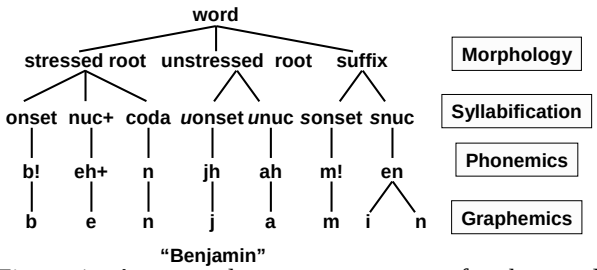

Figure 1: An example ANGIE parse tree for the word "Benjamin."

spelled letters from the spoken utterance, treating the pronounced portion of the word as a generic OOV word. This is followed by an intermediate stage, where the hypotheses of the letter recognition are used to construct a pruned search space for a final sound-to-letter recognizer which directly outputs grapheme sequences. The ANGIE framework serves two important roles simultaneously: specifying the sound/letter mappings and providing language model constraints. The language model is enhanced with a morph *N*-gram, where the morph units are derived via corpus-based techniques. In the following sections, we first describe the ANGIE framework, followed by a detailed description of the multi-pass procedure for computing the spelling and pronunciation of the word from a waveform.

3.1 ANGIE Sound-to-Letter Framework

ANGIE is a hierarchical framework that encodes subword structure using context-free rules and a probability model. When trained, it can predict the sublexical structure of unseen words, based on observations from training data. The framework has previously been applied in bidirectional letter/sound generation (Seneff et al., 1996), OOV detection in speech recognition (Chung, 2000), and phonological modeling (Seneff and Wang, 2002).

A parsing algorithm in ANGIE produces regular parse trees that comprise four distinct layers, capturing linguistic patterns pertaining to morphology, syllabification, phonemics and graphemics. An example parse for the word "Benjamin" is given in Figure 1. Encoded at the pre-terminal-to-terminal layers are letter-sound mappings. The grammar is specified through context-free rules; context dependencies are captured through a superimposed probability model. The adopted model is motivated by the need for a balance between sufficient context constraint and potential sparse data problems from a finite observation space. It is also desirable for the model to be locally computable, for practical reasons associated with the goal of attaching the learned probabilities to the arcs in a finite state network. Given these considerations, the probability formulation that has been developed for ANGIE can be written as follows:

$$P(C_i|C_{i-1}) = P(a_{i,0}|C_{i-1}) \prod_{j=1}^{N-1} P(a_{i,j}|a_{i,j-1}, a_{i-1,j})$$

where C_i is the i^{th} column in the parse tree and $C_i = \{a_{i,j}, 0 \leq j < N\}$, and $a_{i,j}$ is the label at the j^{th} row of the i^{th} column in the two-dimensional parse grid. N is the total number of layers in the parse tree. i and j start at the bottom left corner of the parse tree. In other words, each letter is predicted based on the entire preceding column, and the column probability is built bottom-up based on a trigram model, considering both the child and the left sibling in the grid. The probabilities are trained by tabulating counts in a corpus of parsed sentences.

After training, the ANGIE models can be converted into a finite state transducer (FST) representation, via an algorithm developed in (Chung, 2000). The FST compactly represents sound-to-letter mappings, with weights on the arcs encoding mapping probabilities along with subword structure. In essence, it can be considered as a bigram model on units identified as vertical *columns* of the parse tree. Each unit is associated with a grapheme and a phoneme pronunciation, enriched with other contextual factors such as morpho-syllabic properties. The FST output probabilities, extracted from the ANGIE parse, represent bigram probabilities of a column sequence. While efficient and suitable for recognition search, this *column bigram* FST preserves the ability to generalize to OOV data from observations made at training. That is, despite having been trained on a finite corpus, it is capable of creatively licensing OOV words with non-zero probabilities.

In this work, the probability model was trained on a lexicon of proper nouns, containing both first and last names. During the initial lexical acquisition phase, over 75,000 entries were added to the lexicon via an automatic procedure. Because this yielded many errors, manual corrections have been made, and are ongoing. In a second phase, a further 25,000 names are automatically added to the lexicon, using a two-step procedure. First, the grammar is trained on the original 75,000 words, then using the trained grammar, ANGIE is used to parse the additional 25,000 new names. These parses are immediately added to the full lexicon. Despite generating many erroneous parses, performance improved with the additional training data. After training on the total 100,000 words, the column bigram FST is highly compact, containing around 2100 states and 25,000 arcs. In total, there are 214 unique graphemes (some of which are doubletons such as "th") and 116 unique phoneme units.

3.2 Multi-Stage Speak and Spell Recognition

The multi-stage speak and spell approach is tailored to accommodate utterances with a spoken name followed by the spelling of the name. As depicted in Figure 2, there are three stages: the first is a letter recognizer with an unknown word model, outputting a reduced search space favored by the letter hypotheses; the second pass com-

piles the language models and sound-to-letter mappings into the reduced search space; a final pass uses the scores and search space defined in the previous stage to perform recognition on the waveform, simultaneously generating spelling and phonemic sequences on the word.

At the core of this approach is the manipulation of FSTs, which permits us to flexibly reconfigure the search space during recognition time. The entire linguistic search space in the recognizer can be represented by a single FST (U) which embeds all the language model probabilities at the arc transitions. Generally, U is represented by a cascade of FST compositions:

$$U = C \circ P \circ L \circ G \tag{1}$$

where C contains diphone label mappings, P applies phonological rules, L maps the lexicon to phonemic pronunciations, and G is the language model. The above compositions can be performed prior to run-time or on the fly.

3.2.1 Letter Recognition

The first stage is a simple letter recognizer augmented with an OOV word model (Bazzi and Glass, 2001), which is designed to absorb the spoken name portion of the waveform. The recognition engine is segment-based, using context-dependent diphone acoustic units (Zue et al., 2000). Trained on general telephone-based data (which do not contain spelled names), the acoustic models contain 71 phonetic units and 1365 diphone classes. Using Bazzi's OOV word modeling scheme, unknown words are represented by variable-length subword units that have been automatically derived. The language model, a letter 4-gram, is trained on a 100,000 name corpus, augmented with an unknown word at the beginning of each sentence. This first stage outputs a lattice in the form of an FST, which contains, at the output labels, an unknown word label for the spoken name part of the utterance and letter hypotheses which are useful for the later stages.

3.2.2 Intermediate Stage

A series of FST operations are performed on the output of the first stage, culminating in an FST that defines a reduced search space and integrates several knowledge sources, for the second recognition pass. Since the waveform consists of the spoken word followed by the spelling, the output FST of this stage is the concatenation of two component FSTs that are responsible for recognizing the two portions of the waveform: a first FST maps phone sequences directly to letters, and a second FST, which supports the spelling component, maps phones to the spelled letters.

The first FST is the most knowledge-intensive because it integrates the first pass hypotheses with their corresponding scores, together with additional language models and ANGIE sound-to-letter mappings. A subword trigram language model is applied to subword units that are automatically derived via a procedure that maximizes mutual information. Similar to work in (Bazzi and Glass, 2001), where subword units are derived from phones, the procedure employed here begins with letters and iteratively combines them to form larger units.

The following describes the step-by-step procedure for generating such a final FST (F) customized for each specific utterance, beginning with an input lattice (I) from the first stage. I preserves the acoustic and language model scores of the first stage.

1. Apply subword language model: I is composed with a subword trigram (T). The trigram is applied early because stronger constraints will prune away improbable sequences, reducing the search space. The composition involves L_T, mapping letter sequences to their respective subword units and $L_T^{'}$, the reverse mapping. This step produces an FST (G_1) with letters at both the inputs and outputs, where
$$G_1 = I \circ L_T \circ T \circ L_T^{'} \tag{2}$$

2. Apply ANGIE model: G_1 is composed with the column bigram FST (A). This requires an intermediate FST (L_A), mapping letter sequences to ANGIE grapheme symbols. The result is G_2, where
$$G_2 = G_1 \circ L_A \circ A \tag{3}$$
G_2 codifies language information from ANGIE, a subword trigram, and restrictions imposed by the letter recognizer. Given a letter sequence, G_2 outputs phonemic hypotheses.

3. Apply phonological rules: The input and output sequences of G_2 are reversed to yield $G_2^{'}$, and we apply
$$F_1 = P \circ G_2^{'} \tag{4}$$
This expands ANGIE phoneme units to allowable phonetic sequences for recognition, in accordance with a set of pronunciation rules, using an algorithm described in (Hetherington, 2001). The resultant FST (F_1) is a pruned lattice that embeds all the necessary language information to generate letter hypotheses from phonetic sequences.

4. Create second half FST: The FST (F_2) necessary for processing the spelling part of the waveform is constructed. This begins by composing I, the FST containing letter hypotheses from the first stage, with an FST (L_B) representing baseforms for the letters, followed by the application of phonological rules, similar to Step 3.
$$F_2 = P \circ L_B \circ I \tag{5}$$

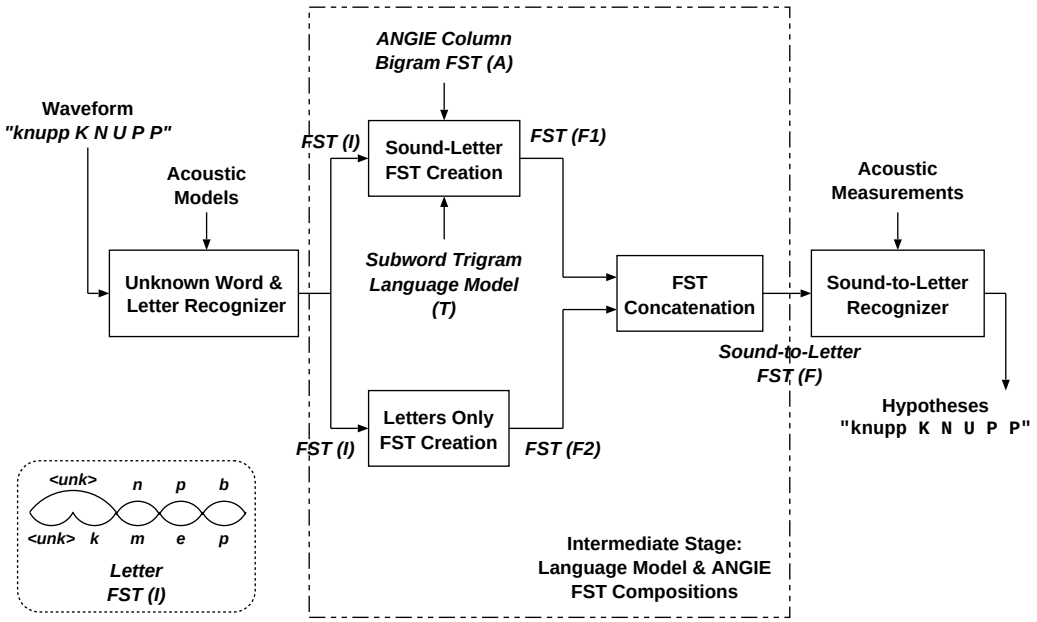

Figure 2: A schematic illustrating the multi-stage speak and spell recognition procedure. I, capturing the first stage letter hypotheses, is input to an intermediate stage, where F_1 (see steps 1 to 3 in Section 3.2.2) is concatenated with F_2 (step 4). The result F defines the search space for the final stage.

5. Concatenate two parts: The final FST (F) is created by concatenating the FSTs corresponding with the first (F_1) and second (F_2) portions of the speak and spell waveform.

$$F = F_1 \cdot F_2 \tag{6}$$

As described above, F_1 is particularly rich in knowledge constraints, because all the scores of the first stage are preserved. These are acoustic and language model scores associated with those hypotheses, determined from the spelled part of the waveform. Hence F_1 contains hypotheses that are favored by the language and acoustics scores in the letter recognition pass, to be applied to the spoken part of the waveform in the next pass. The scores are enriched with an additional subword trigram and the ANGIE model to select plausible sound-to-letter mappings.

3.2.3 Sound-to-Letter Recognition

The sound-to-letter recognizer conducts an entirely new search, using the enriched language models in a reduced search space, along with the original acoustic measurements from the first pass. Mapping phonetic symbols to letter symbols, the input FST (F) is equivalent to $P \circ L \circ G$, incorporating phonological rules and language constraints. It is then composed on-the-fly with a pre-loaded diphone-to-phone FST (C), thereby completing the search space as defined in Equation 1.

The final letter hypothesis for the name is extracted from the output corresponding to the spoken name portion of the utterance, taken from the highest scoring path. Essentially, this final pass integrates acoustic information from the spelled and spoken portions of the waveform, with language model information from the grapheme-phoneme mappings and the morph *N*-gram.

3.2.4 Phoneme Extraction

Phoneme extraction is performed using an additional pass through the search engine of the recognizer. In the ORION system, the phoneme sequence is only computed after the user has confirmed the correct spelling. The procedure is analogous to the sound-to-letter process described above, except that, instead of using output from the first-stage letter recognizer, a single letter sequence constrains the search. The sequence may either be the answer as confirmed by the user during dialogue, or the highest scoring letter sequence output from the sound-to-letter recognizer. A series of FST compositions is performed to create an FST that can compute a phonemic sequence in accordance with ANGIE model mappings, associated with the given letter sequence and the acoustic waveform. Again, the FST contains two portions, for processing each half of the speak and spell waveform. The first applies ANGIE to map phonetic symbols to phonemic symbols, restricted to paths that correspond with the input letter sequence. The second half supports the spelled letter sequence. Following FST creation, the final FST is uploaded to the search engine, which conducts a new search using the FST and the original acoustic measurements. The phoneme sequence for the name is taken as the output from the highest scoring path corresponding with the spoken part of the waveform.

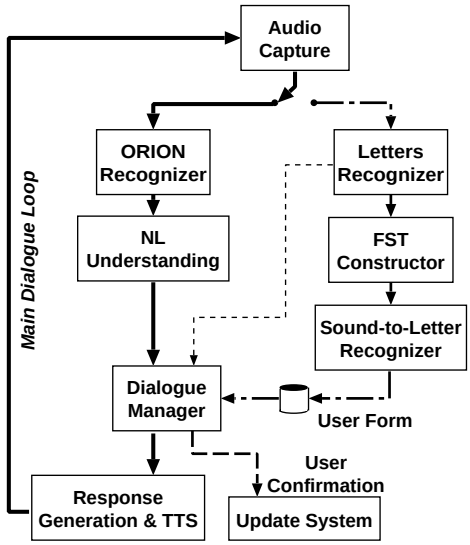

Figure 3: A schematic depicting the multi-threaded implementation for the speak and spell system.

4 System Integration

Our integration experiments were conducted in the ORION system, which is based on the GALAXY Communicator architecture (Seneff et al., 1998). In GALAXY, a central hub controls a suite of specialized servers, where interaction is specified via a "hub program" written in a scripting language.

In order to carry out all of the activities required to specify, confirm, and commit new words to the system's working knowledge, several augmentations were required to the pre-existing ORION system. To facilitate the automatic new word acquisition process, two new servers have been introduced: the FST constructor and the system update server. The role of the FST constructor is to perform the series of FST compositions to build FST F as described previously. Via rules in the hub program, the constructor processes the output of the letter recognizer to derive an FST that becomes input to the final sound-to-letter recognizer.

The second new server introduced here is the system update server, which comes into play once the user has confirmed the spelling of both their first and last names. At this point, the NL server is informed of the new word additions. It has the capability to update its trained grammar both for internal and external use. It also creates a new lexicon and class n-gram for the recognizer.

In addition to the NL update, the recognizer also needs to incorporate the new words into its search space. At present, we are approaching this problem by recompiling and reloading the recognizer's search FSTs. In the future, we plan to augment the recognizer to support incremental update of the lexical and language models. The system update server is tasked with re-generating the FSTs asynchronously, which are then automatically reloaded by the recognizer. Both the recognizer and the NL system are now capable of processing the newly specified name, a capability that will not be needed until the next time the new user calls the system.

One interesting aspect of the implementation for the above processing is that the system is able to make use of parallel threads so that the user does not experience delays while their name is being processed through the multiple stages. Figure 3 illustrates a block diagram of the dialogue flow. The letter recognizer processes the user's first name during the main recognition cycle of the turn. Subsequently, a parallel second thread is launched, in which the second stage recognizer searches the FST created by the FST constructor as described previously. In the mean time, the main hub program continues the dialogue with the user, asking for information such as contact phone numbers and email address. The user's last name is processed similarly. At the end of the dialogue, the system confirms the two names with the user. If they are verified, a system update is launched, while the system continues the dialogue with the user, perhaps enrolling their first task. If the user rejects a proposed spelling, the system will prompt them for a keypad entry of the name (Chung and Seneff, 2002), which will provide additional constraints.

5 Experiments

Experiments have been conducted to evaluate the ability to acquire spellings and pronunciations for an open set of names. We have selected a test set that combines utterances from a preliminary ORION data collection during new user enrollment and previous utterances collected from the JUPITER system (Zue et al., 2000), where at the beginning of each phone call, users are asked to speak and spell their names in a single utterance.

Thus far, 80% of the test set comes from JUPITER data, in which users mostly provided first names. However, the trained models are designed to support both first and last names. As yet, no attempts have been made to separately model first and last names.

Two test sets are used for evaluation. Test Set A contains words that are present in ANGIE's 100K training vocabulary with 416 items of which 387 are unique; Test Set B contains words that are previously unseen in any of the training data, with 219 items of which 157 are unique. These test sets have been screened as best as possible to ensure that the spelled component corresponds to the spoken name in the utterance.

5.1 Results and Discussion

For each test set, letter error rates (LER) and word error rates (WER) are computed for the output for the first letter recognizer, and the output for the entire multi-stage

	LER (%)	*WER (%)*
Stage 1 Output	12.8	40.4
Multi-Stage System I	8.3	25.7
Multi-Stage System II	8.4	27.4

Table 1: *Letter Error Rates (LER) and Word Error Rates (WER) for Test Set A, containing 416 words that are in the* ANGIE *training vocabulary. System II lacks the morph trigram.*

	LER (%)	*WER (%)*
Stage 1 Output	19.1	58.9
Multi-Stage System I	14.3	48.9
Multi-Stage System II	12.4	46.1

Table 2: *Letter Error Rates (LER) and Word Error Rates (WER) for Test Set B, containing 219 words that are previously unseen. System II lacks the morph trigram.*

system (Multi-Stage System I). In an additional experiment, the subword trigram is omitted in the intermediate stage (Multi-Stage System II). Results are summarized in Tables 1 and 2.

When evaluating output from the first-stage letter recognizer only, it is found that errors remain high (40.4% WER for Test Set A and 58.9% WER for Test Set B). It should be noted that none of the training data for the acoustic models contain any letter spellings, contributing to relatively poor performance compared to that in other domains using the same models. Many of the errors are also caused by poor detection of the transition from the spoken name to the spelled portion of the waveform. Deletions occur when part of the spelled portion is mistakenly identified as part of the unknown word or insertions arise when the end of a spoken word is confused for a spelled letter. However, the multi-stage system produces a marked improvement if we compare it with the single-stage letter recognizer as a baseline. WER improves by 36.4% (from 40.4% to 25.7%) for Test Set A, and 17.0% (from 58.9% to 48.9%) for Test Set B. The improvement is more pronounced for Test Set A because the words have been observed in the ANGIE training data. The most commonly confusable letter pairs are: *M/N, A/E, J/G, Y/I, L/O, D/T*. These letters are confusable both acoustically in the spelled letters as well as in the pronunciation of the spoken word.

When the subword trigram is removed from the language model in the later stages, further WER improvements result in Test Set B (46.1%), although performance in Test Set A deteriorates. We infer that unknown words benefit more with a less constrained language model, and when more weighting is given to the ANGIE model for generating possible spelling alternatives.

PER (%)	Overall	Correct Spelling	Incorrect Spelling
Test Set A	15.0	6.2 (309)	40.9 (107)
Test Set B	25.5	11.0 (112)	40.6 (107)

Table 3: *Phoneme Error Rates (PER) computed for two test sets. In Test Set A, 309 words (74.3%) are spelled correctly, and 107 words (25.7%) are incorrect. In Test Set B, 112 (51.1%) words are correctly spelled.*

To evaluate the phoneme extraction accuracy, the best letter hypothesis of the multi-stage system is used to compute the phonemes, as described in Section 3.2.4. In the actual ORION system, when a user confirms the correct spelling of their name, if the name exists in the training pronunciation lexicon, the phoneme extraction stage may be redundant. This assumes the pronunciation lexicon itself is reliable, and contains all the correct alternate pronunciations of the word. For the purpose of evaluation, we examine the phoneme outputs of both in-vocabulary Test Set A, and OOV Test Set B, whose phonemic baseforms have been hand-transcribed.

Within ANGIE, phonemes are marked for lexical stress and syllable onset positions. There are also many special compound phonemic units (e.g., */sp, sk, st/*). A much smaller phoneme set of 50 units is derived for evaluation, by applying rules to collapse the phoneme hypotheses. The phoneme error rate (PER) for Test Set A and B are depicted in Table 3. Error rates are provided for the subsets of words where the letter hypotheses are either correct or incorrect. Many of the confusable phoneme pairs are vowels: *ih/iy, ae/aa, eh/ey.* Other commonly confused phoneme pairs are: *m/n, en/n, er/r, l/ow, d/t, s/z, th/dh.*

In another experiment, we evaluated the accuracy of the phoneme extraction by using the correct letter sequence as input, instead of the highest scoring letter sequence. The PER for Test Set A is 7.2% and the PER for Test Set B is 13.3%. While phoneme error rates are generally higher than letter error rates, it should be noted that the reference baseforms for the names contain only one or two alternate pronunciations for each name. However, it is not uncommon for a name to have many irregular pronunciation variants, which are not covered in the reference baseforms. Also the phonemic baseform determined by the recognizer is likely to be one preferred by the system for the particular speaker, assumed to be the owner, of the name. Therefore, we believe that the baseforms favored by the system may be more appropriate for subsequent recognition, especially if the name is to be spoken by the same speaker. This may be the case in spite of the mismatch between the favored phonemic baseform and that in the pronunciation dictionary.

6 Conclusions and Future Work

This paper has described a methodology and implementation for automatically acquiring user names in the ORION task delegation system. It has been shown that a novel multi-stage recognition procedure can handle an open set of names, given waveforms with the spoken name followed by the spelled letters. The overall system is also capable of incorporating the new name immediately into its language and lexical models, following the dialogue.

Future work is needed on many parts of the system. As more data are collected, future experiments will be conducted with larger test sets. We can improve the letter recognizer by explicitly modeling the transition between the unknown word and the spelling component. For instance, by adding prosodic features we may be able to improve the detection of the onset of the spelling part.

Our final selection process is based only on the proposed spellings obtained from the pronounced word, after feeding information from the spelled part into the second stage. However, performance may improve if we apply a strict constraint during the search, explicitly allowing only paths where the spoken and spelled part of the waveforms agree on the name spelling. Alternatively, a *length* constraint can be imposed on the letter sequence, once it has been observed that the second stage hypotheses for the spoken and the spelled components are inconsistent in length.

As an unconstrained name recognizer, the system described here handles in the same way both observed data and previously unseen data. We would like to experiment with adding a parallel component that explicitly models some in-vocabulary words. This may boost overall accuracy by lexicalizing the most common names, such that only words that are identified as OOV need to be processed by the ANGIE sound-to-letter stage.

In regards to implementation, the current hub-server configuration has allowed us to rapidly implement the system and conduct experiments. The multi-threaded approach, implemented using the hub scripting language, has been effective in allowing a smooth dialogue to proceed while the multi-stage processing takes place in the background. However, we anticipate that the multi-stage approach can be improved by folding all three stages into a single recognition server, eventually allowing real-time operation. In this case, multi-threading would only be needed for the final stage that incorporates the new words into the on-line system.

The long-term objective of this work is to learn the pronunciations and spellings of general OOV data in spoken dialogue systems on domains where OOV proper nouns are prevalent. Future experiments will involve general classes of unknown words such as names of geographical locations or businesses.

References

Bazzi I. and Glass J. 2001. "Learning Units for Domain-Independent Out-of-Vocabulary Word Modeling," *Proc. Eurospeech*, 61–64, Aalborg, Denmark.

Chung G. 2000. "A Three Stage Solution for Flexible Vocabulary Speech Understanding," *Proc. ICSLP*, Beijing, China.

Chung G. and Seneff S. 2002. "Integrating Speech with Keypad Input for Automatic Entry of Spelling and Pronunciation of New Words," *Proc. ICSLP*, 2053–2056, Denver, CO.

Damper R. I. et al. 1998. "Comparative evaluation of letter-to-sound conversion techniques for English text-to-speech synthesis," *Proc. IWSS*, 53–58, Jenolan Caves, Australia.

Decadt B. et al. 2002. "Transcription of Out-of-Vocabulary Words in Large Vocabulary Speech Recognition Based on Phoneme-to-Grapheme Conversion," *Proc. ICASSP*, Orlando, FL.

Galescu L. and Allen J. 2002. "Name Pronunciation with a Joint N-Gram Model for Bi-directional Grapheme-to-Phoneme Conversion," *Proc. ICSLP*, 109–112, Denver, CO.

Hetherington I. L. 2001. "An Efficient Implementation of Phonological rules using Finite-State Transducers," *Proc. Eurospeech*, Aalborg, Denmark.

Font Llitjos A. and Black A. 2001. "Knowledge of Language Origin Improves Pronunciation Accuracy of Proper Names," *Proc. Eurospeech*, Aalborg, Denmark.

Marchand Y. and Damper R. I. 2000. "A multi-strategy approach to improving pronunciation by analogy," *Computational Linguistics*, 26(2), 195–219.

Meng H. et al. 1996. "Reversible Letter-to-sound / Sound-to-letter Generation Based on Parsing Word Morphology," *Speech Communication*, 18(1), 47–64.

Ngan J. et al. 1998. "Improved Surname Pronunciations Using Decision Trees," *Proc. ICSLP '98*, Sydney, Australia.

Rentzepopoulos P. and Kokkinakis G. K. 1996. "Efficient Multilingual Phoneme-to-Grapheme Conversion Based on HMM," *Computational Linguistics*, 22(3), Sept.

Schillo C. et al. 2000. "Grapheme Based Speech Recognition for Large Vocabularies," *Proc. ICSLP*, Beijing, China.

Seneff S. et al. 1996. "ANGIE: A new framework for speech analysis based on morpho-phonological modeling," *Proc. ICSLP*, 110–113, Philadelphia, PA.

Seneff S. et al. 1998. "Galaxy-II: A Reference Architecture for Conversational System Development," *Proc. ICSLP*, 931–934, Sydney, Australia.

Seneff S. et al. 2000. "Orion: From On-line Interaction to Off-line Delegation," *Proc. ICSLP*, Beijing China.

Seneff S. and Wang C. 2002. "Modeling Phonological Rules through Linguistic Hierarchies," *Proc. ISCA Workshop*, pp. 71–76, Estes Park, Colorado.

Zue V. et al. 2000. "JUPITER: A Telephone-Based Conversational Interface for Weather Information," *IEEE Transactions on Speech and Audio Processing*, Vol 8, No. 1, Jan.