INTEGRATING SPEECH WITH KEYPAD INPUT FOR AUTOMATIC ENTRY OF SPELLING AND PRONUNCIATION OF NEW WORDS

Grace Chung

Corporation for National
Research Initiatives
Reston, VA 20191

Stephanie Seneff[*]

Spoken Language Systems Group
MIT Laboratory for Computer Science
Cambridge, MA 02139

ABSTRACT

This paper describes research whose ultimate aim is to support automatic entry of new words into a spoken dialogue system through interaction with a user. This research demonstrates an important step towards this goal, through a procedure which integrates information made available via the telephone keypad with a spoken instance of the target word, to produce a candidate spelling and pronunciation for the word. Through the use of a parsing mechanism applied to a 73,000 word proper name lexicon [4], we have been able to create a finite-state transducer (FST) that maps phonetics to graphemics, which can be composed with an FST derived from the keypad input to greatly reduce the search space. Experiments conducted on both the OGI name corpus [2] and a set of enrollment data obtained from our Mercury system [5] validate the procedure.

1. INTRODUCTION

For some time, we have been involved with research in handling out-of-vocabulary (OOV) words that arise in conversations with spoken dialogue systems. The OOV words are often proper nouns, which are tremendously difficult to predict. The set of people's names, originating from all languages, is so vast that the vocabularies of word and subword-based recognizers generally do not afford reasonable coverage or performance. For a typical application requiring the enrollment of a new user, where the user's name is unknown *a priori*, it is both impractical and inadequate to attempt full coverage by using a very large vocabulary recognizer. We present an alternative method here which offers a more efficient and flexible solution. This approach aims to use knowledge gathered from trained subword models to predict the new user name, without assuming a fixed vocabulary.

We envision a system that can learn new words by interacting with a human user, for example, by prompting the user to verbally spell the word, repeat the pronunciation, and verify the system's best guess, with a follow-up keypad request if the hypothesis is incorrect. This is particularly relevant to new user enrollment, necessary in many of our systems [5, 6]. In our current model, users interact at a Web page, typing in their information, which is later entered manually by experts into the recognizer and linguistic components. As our systems become more widely available, it becomes important to eliminate this step of manual intervention, thus empowering users to fully enter their own enrollment information. Furthermore, this vision is applicable to any scenario involving new-word acquisition.

In this paper, we describe some experiments that simulate an automated enrollment system, where the system generates a pronunciation and spelling for the new user's name. Users, who call the system on the phone, would be asked to enter the spelling of their names on a telephone keypad, and speak their full name. We have assembled a recognizer, which directly hypothesizes grapheme sequences from phones, along with underlying phonemic baseforms. The language models, derived from subword structure and grapheme information, are not tied to a fixed vocabulary. While models are trained on a name lexicon, they also support a much larger set of *previously unseen* words due to their ability to extrapolate from training probabilities. This methodology has the capability to exploit language constraints from subword information including spelling, and in the process, directly hypothesize spellings from the phone sequences. It also provides the opportunity to maximally harness information from user-entered keypad sequences; that is, the keyed-in sequences are directly applied to constrain phonetic hypotheses top-down, before search commences. This is accomplished using a framework of weighted FSTs. An FST imposes the ensemble of constraints enforced by the keypad entries and language model at the very beginning of the search, greatly shrinking the search space.

It is our hope that such integrated constraints will ultimately boost accuracy to a reliable level whereby a function such as user enrollment can be conducted smoothly without a GUI, and effectively, recognizer models can be dynamically updated with the new word. In the next sections, we will detail our sound-to-letter name recognizer, and explicate the algorithms for creating the language models and instantiating the FSTs. We discuss some exploratory experiments using the keypad and the linguistic constraints, and provide some results for recognition experiments conducted on test sets containing people's names.

2. APPROACH

Proper names pose a great challenge for both recognition and text-to-speech because the space of unique names is vast, and the letter-to-sound rules are highly variable. It is equally difficult to incorporate adequate linguistic constraints for the recognizer search as it is to predict pronunciations from letters (and vice versa.) Our approach utilizes a sound-to-letter recognizer that hypothesizes

[*] The research at CNRI was supported by DARPA under contract N66001-00-2-8922 monitored through SPAWAR Systems Center – San Diego. The research at MIT was supported by DARPA under contract N66001-99-1-8904 monitored through Naval Command, Control and Ocean Surveillance Center, and under contract number NBCH1020002 monitored through the Dept. of the Interior.

In *Proceedings of the 7th International Conference on Spoken Language Processing*, Sep. 16-20, 2002, Denver, Colorado, pp. 2061–2064.

grapheme sequences from phones. The development of this recognizer involves two phases. During the first phase, we use the ANGIE framework [4] to compile a grammar that codifies subword structure information and sound-to-letter mappings, learned from a corpus of proper nouns. The ANGIE grammar is then converted into a compact FST, with letter-to-phoneme mappings. In the second phase, we develop a method for incorporating several knowledge sources into the language model. The key to this process is the use of FSTs to integrate the language constraints: the ANGIE model, a subword bigram and keypad information. Since keypad information and probability models are combined and optimized before the recognition phase, the search space is greatly pruned, benefiting both performance and efficiency.

2.1. ANGIE Proper Noun Modeling

ANGIE is a hierarchical framework that encodes subword structure using context-free rules and a probability model. When trained, it has the ability to predict and score the sublexical structure of unseen words based on observations from training data. Our past work has applied ANGIE in OOV detection in recognition [1] and bi-directional letter/sound generation [4]. Presently, we have combined our previous approaches by integrating grapheme-phoneme conversion capabilities with the recognition search.

The parsing algorithm in ANGIE produces a regular parse tree that comprises 4 layers below the word level. Previously, we have experimented with the functionality and context modeling encoded at each layer. The distinct layers capture linguistic patterns pertaining to morphology, syllabification, phonemics and graphemics. More importantly, the pre-terminal-to-terminal layers express letter-to-sound mappings. The probability model is trained from a lexicon, which is represented in two tiers: words are defined in terms of morph sequences while morphs are defined by phonemic baseforms. These morphs are syllable-sized units enhanced with contextual information such as spelling, stress, and word position.

A critical aspect of using ANGIE for modelling letter-to-sound mappings is the acquisition of a suitable lexicon for training. To this end, we have acquired a lexicon of over 73,000 proper nouns, representing both first and last names. The original lexical representations are obtained through an automatic procedure, but the lexical entries contains many errors, which we have been manually correcting, through an ongoing labor-intensive process. The morph lexicon currently contains over 14,000 unique entries. On a development set of 4904 words, not present in the training data, the per letter perplexity is measured to be 18.6. This is an upper bound measure since a portion of the probability space is lost to alternative parses.

The trained ANGIE grammar can be converted into a *column bigram* FST format, as previously described in [1], which is a compact representation of the search space covered by ANGIE's models. 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, the 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. For our training set, the column bigram FST consists of 1888 arcs and 19,072 states. There are 214 ANGIE graphemes (some of which are letter doubletons such as "th") and 116 ANGIE phonemes, augmented with features such as onset position and stress.

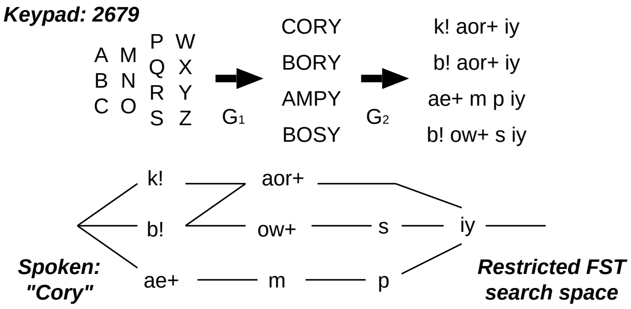

Fig. 1. A schematic for integrating keypad input with phonetic recognition, to produce a hypothesized spelling and pronunciation for the name "Cory." Entered at the keypad is the sequence "2679," producing a total of 144 possible four-letter sequences. Using a subword language model, FST G_1 sets out probable names, and FST G_2 employs ANGIE to output likely pronunciations.

2.2. FST Recognizer

The recognition engine is a segment-based FST recognizer, using context-dependent diphone acoustic units [7]. In the acoustic models, there are 71 phonetic units and 1365 diphone classes. Central to the search algorithm is the use of FSTs. A single FST (U) defines the entire linguistic search space, embedding language model probabilities at the arc transitions. Generally, U is represented by a cascade of FST compositions: $U = C \circ P \circ L \circ G$, where C contains context-dependent label mappings, P applies phonological rules, L maps the lexicon to phonemic pronunciations and G is the language model. The recognizer supports the uploading of a single pre-composed FST prior to recognition or on-the-fly composition of separate FSTs at recognition time. For on-the-fly composition, a new customized FST lattice can be uploaded at the beginning of every new utterance. This will be our mode of operation. In fact, our configuration uploads an FST (F), one that is equivalent to $P \circ L \circ G$. At the beginning of processing each waveform, F is composed, optimized, then uploaded, to be combined on-the-fly with a simple diphone-to-phone FST (C), which resides permanently within the recognition engine.

For our sound-to-letter configuration, F equates to an FST that transduces phones to letters, embedded with probability scores derived from a number of sources. The search space and the probabilities are defined by an integrated set of constraints, enforced by the keypad entries, a subword bigram, the ANGIE column bigram FST, and phonological rules on phonetic sequences. This process is illustrated schematically in Figure 1.

The subword bigram is intended to provide longer-distance language constraints than the ANGIE FST. Yet, it is also designed to be quite loosely constrained, in order to support previously unknown sequences. To this end, we have devised a set of syllable-sized units, seeded from morphs in the ANGIE lexicon. The original morphs are stripped of all stress, position and pronunciation information so that all morphs with identical spellings are blended together, creating a highly generalized language model of 8049 units, with an average length of 4 letters. A perplexity measure of 88 was obtained from the training set.

The procedure to create the utterance-specific F to support the phonetic search constrained by the keypad input, as well as providing the mapping from phones to letters, is composed of a series of steps as follows:

1. FST composition begins when the user has keyed in her name at the keypad. This phase results in a keypad FST (K), wherein, for each key pressed, a choice of three (or, rarely, four) letters are possible. For a name keyed in with length n letters, the FST consists of $n + 1$ states in which each state has 3 or 4 transitions to the next state in the sequence. Each arc is labeled with a possible letter. Hence, there are at least 3^n possible spellings for each keyed-in name.

2. K is composed with a subword bigram (B). The bigram is applied early because stronger bigram constraints will serve to prune away improbable sequences, reducing the search space. The composition is achieved with an intermediate FST (L_B), that maps letter sequences to their respective morphs. This step produces an FST (G_1), where

$$G_1 = K \circ L_B \circ B \tag{1}$$

3. To prepare for the next phase, the morph labels are discarded and letter symbols are projected to both the FST inputs and outputs. At this point, G_1 is also pruned significantly. It has been empirically determined that a best-first search, producing the top $N = 50$ paths, is sufficient.

4. The pruned 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

$$G_2 = G_1 \circ L_A \circ A \tag{2}$$

   G_2 codifies language information from ANGIE, a subword bigram, and restrictions imposed by the keypad entries. Given a letter sequence, G_2 outputs phonemic hypotheses.

5. The next stage is to apply phonological rules. The input and output sequences of G_2 are reversed to yield $G_2^{'}$, and we apply

$$F = P \circ G_2^{'} \tag{3}$$

   This will expand ANGIE phoneme units to allowable phonetic sequences for recognition, in accordance with a set of pronunciation rules. The algorithm employed here is described in [3]. The resultant FST (F) is a pruned lattice that embeds all the necessary language information to generate letter hypotheses from phonetic sequences.

The underlying phonetic pronunciations of our recognized output can be recovered by retrieving the top-scoring phonetic sequence that corresponded with the best letter hypothesis.

3. EXPERIMENTS

Experiments are performed on two sets of data: (1) speech collected over the past eight months for users logging onto the Mercury flight reservation system [5], and (2) speech from the telephone based OGI Spelled and Spoken Word Corpus [2].

In Mercury, users are asked to say their full name in one turn; many of the users also opted to spell their names. We have gathered a test set of 395 utterances; this contains a total of 807 words, of which 205 are unique. Since the lexicon used to train ANGIE is gathered from a separate process, a considerable number of words (16.0%) from this set are absent from the training lexicon. These are considered to be OOV. Each utterance contains, consecutively, the speaker's first and last name, with some including their middle name. Most of these words are continuously spoken, without deliberate pauses in between. All utterances with spellings are excluded at this time.

For the OGI Corpus, we tested on isolated single-word first and last names, from the "say-lname" and "say-fname" sets. Omitting utterances with spurious speech and spelling, the test set contains 4291 utterances where 61.6% are surnames. These are extracted from 3009 separate calls. In all, the OOV rate with respect to the ANGIE training lexicon is 16.2%; there are 687 unique first names and 1965 unique surnames.

3.1. Telephone Keypad

Prior to conducting recognition experiments, we investigate the quality of the language models in combination with the keypad entries. We pose the question of how well these constraints alone can predict a word.

With each of the 807 words in the Mercury test set, we computed the FST G_2 as described by Equation 2. G_2 encompasses the column bigram, the subword bigram and the keypad constraints. The highest scoring path in G_2 represents the best guess for the word given only the language model and keypad, with no acoustic information. For the 807 words, we evaluate how often the correct word is hypothesized. The results yield a 13.9% letter error rate (LER) and a 43.1% word error rate (WER). For the purpose of comparison, an alternative FST using a letter trigram combined with keypad entries is created, for each word. This FST is constructed in much the same fashion as described in Section 2.2, except that the only language model is a simple letter trigram. Similarly, we evaluate the quality of the best scoring path in each FST. For this system, the performance dropped to 25.3% LER and 72.9% WER. We can conclude that the column bigram teamed with the subword bigram creates a substantially more powerful language model than a letter trigram. However, it is also apparent that keypad and language information alone are poor predictors of these names.

We are also interested in how often the correct word is missing from the graph contained in G_2. It is found that the correct answer is missing in 5.08% of the graphs. While this constitutes an upper bound on the WER in recognition, it is still possible that the hypothesized spelling may be reasonable approximations of erroneous words.

3.2. Recognition

Recognition experiments are performed on both the Mercury and OGI test sets. Probabilities in the ANGIE column bigram are scaled by 0.5, a value that has been determined on a development set.

In order to handle the full-name contents of the Mercury utterances, we simulate a user entering their entire spelling at the telephone keypad, which could be done, for example, in separate turns prompted by the system. To facilitate this, after the keys are entered, G_2, from Equation 2, is synthesized corresponding to

Test Set	*LER (%)*	*WER (%)*	*SER (%)*
Mercury (395 utts)	3.4	13.5	24.6
OGI (4291 utts)	3.9	16.1	16.1

Table 1. *Letter Error Rates (LER) and Word Error Rates (WER) for the Mercury and OGI test sets. Both sets have OOV rates of around 16%.*

Test Set	IV subset (84%)		OOV subset (16%)	
	LER (%)	*WER (%)*	*LER (%)*	*WER (%)*
Mercury	1.7	8.1	12.0	43.2
OGI	1.8	8.1	13.3	57.3

Table 2. *Letter Error Rates (LER) and Word Error Rates (WER) for the In-Vocabulary (IV) and Out-of-Vocabulary (OOV) portion of the Mercury and OGI test sets.*

each word spoken. The resulting FSTs are then concatenated in the order of the words spoken. Subsequent to this, phonological rules are applied to produce F, as in Equation 3. This method not only enforces the number of letters in each word but also strictly the number of words and placement of word boundaries.

3.2.1. Results

Error rates for the two test sets are tabulated in Table 1. For the Mercury test set, a LER of 3.4% is achieved. The WER is 13.5%, and sentence[1] error rate is 24.6%. The OGI test set, with a richer diversity of names, attains 3.9% LER and 16.1% WER. The in-vocabulary (IV) and OOV portions of the test sets are analyzed separately, as seen in Table 2. Both sets perform substantially better on IV words (8.1% WER), than on OOV data, where Mercury data yields 43.2% WER and the OGI set degrades to 57.3% WER. However, our OGI test set is composed of predominantly surnames, which are much harder (4.8% LER and 20.2% WER) whereas the first names are substantially easier (2.3% LER and 9.5% WER). For the N-best recognition outputs of the Mercury data, only 5.08% of the words do not appear in the list, at $N = 50$. This figure is 6.2% in the OGI set. From the Mercury set, it appears that, for words whose correct answers do exist within the lattice F, the correct answers all emerge at the top 50 recognition output. All the words whose correct answers have failed to surface are in the OOV set.

For the Mercury data, we also perform a preliminary analysis of the proposed pronunciations, and find that 73% of the words have reasonable pronunciations, with the majority of the errors being attributable to erroneous spellings. Many of the errors for correctly spelled words involve stress assignment (*FIL*isko vs fil*IS*ko). Common spelling errors are m/n and u/v confusions, which are reasonable acoustic errors.

4. DISCUSSION

While the above recognition results are very promising, we believe that further improvements will be possible, mainly through enhancements in the quality of the lexicon. In comparing the keypad-only experiment with the recognition results, it is evident that the integration with the acoustic information is pivotal in extracting the right answer.

We are also encouraged by performance on the OOV subset. OOV rates of around 16% may typically be expected for a general purpose enrollment application. Almost half of these unknown words are salvaged with 100% correct spelling for both test sets. In principle, the remaining errorful words could be recovered semi-automatically by a human inspecting the N-best list, which contains the correct answers for up to 94.9% of the test words. Alternatively, the system could search a known database of names for the best matching spelling. Overall results seem to indicate that our proposed schema for combining a flexible vocabulary sound-to-letter recognizer with keypad entries can be a viable solution.

5. FUTURE WORK

This paper has presented a new system that allows users to verbally enroll their full names, with the aid of telephone keypad entry. We hope more performance improvements can be gained as we optimize the ANGIE grammar, which may in the future be used as a stand-alone letter-to-sound module. We intend to increase the coverage of this grammar, and experiment with a more extensive set of proper names. Moreover, we will also address (1) recognizing spelled letters, as an optional means for augmenting the user enrollment process, and (2) recognizing unknown proper nouns embedded within a complete sentence. Ultimately, combining keypad entry and spelled letters with the spoken word may become a practical way to verbally add any new words into the recognizer.

6. ACKNOWLEDGEMENTS

The authors are grateful to T.J. Hazen and Lee Hetherington for supplying the FST tools and recognizer search implementation.

7. REFERENCES

[1] G. Chung, “A Three-Stage Solution to Flexible Vocabulary Understanding,” *Proc. ICSLP ’00*, Beijing, China, Oct. 2000.

[2] R. Cole et al., “A Telephone Speech Database of Spelled and Spoken Names,” *Proc. ICSLP ’92*, Banff, Canada, Oct. 1992.

[3] I. L. Hetherington, “An Efficient Implementation of Phonological rules using Finite-State Transducers,” *Proc. Eurospeech ’01*, Aalborg, Denmark, Sept. 2001.

[4] S. Seneff, et al., “ANGIE: A new framework for speech analysis based on morpho-phonological modelling,” *Proc. ICSLP ’96*, Philadelphia, PA, Oct. 1996.

[5] S. Seneff and J. Polifroni, “Dialogue Management in the Mercury Flight Reservation System,” *Proc ANLP-NAACL ’00*, Seattle, WA, April, 2000.

[6] S. Seneff, et al. “Orion: From On-line Interaction to Off-line Delegation,” *Proc. ICSLP ’00*, Beijing China, Oct. 2000.

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

[1] The sequence of first-name, last-name, for instance.