Empowering End Users to Personalize Dialogue Systems through Spoken Interaction[1]

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

†Spoken Language Systems Group
MIT Laboratory for Computer Science
Cambridge, Massachusetts USA
{seneff, wangc}@sls.lcs.mit.edu

‡Corporation for National Research Initiatives
Reston, Virginia, USA
gchung@cnri.reston.va.us

Abstract

This paper describes recent advances we have made towards the goal of empowering end users to automatically expand the knowledge base of a dialogue system through spoken interaction, in order to personalize it to their individual needs. We describe techniques used to incrementally reconfigure a preloaded trained natural language grammar, as well as the lexicon and language models for the speech recognition system. We also report on advances in the technology to integrate a spoken pronunciation with a spoken spelling, in order to improve spelling accuracy. While the original algorithm was designed for a "speak and spell" input mode, we have shown here that the same methods can be applied to separately uttered spoken and spelled forms of the word. By concatenating the two waveforms, we can take advantage of the mutual constraints realized in an integrated composite FST. Using an OGI corpus of separately spoken and spelled names, we have demonstrated letter error rates of under 6% for in-vocabulary words and under 11% for words not contained in the training lexicon, a 44% reduction in error rate over that achieved without use of the spoken form. We anticipate applying this technique to unknown words embedded in a larger context, followed by solicited spellings.

1. Introduction

Spoken dialogue systems are emerging as an effective means for humans to access information spaces through natural spoken interaction with computers. A significant enhancement to the usability of such systems would be the automatic acquisition of 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). There has been significant research recently on the topic of automatically learning the meaning of words represented symbolically [4–6], assuming the pronunciation and spelling of the words are provided. Our goals are somewhat different: we assume that the meaning of the new word can be inferred by its surrounding dialogue context; subsequently, the system solicits the spelling through dialogue interaction, and immediately integrates the spoken and spelled forms into all relevant components.

In [2], we described a technology solution to the problem of integrating the information provided by a "speak and spell" utterance, as in "Jane J A N E," and its subsequent incorporation into a spoken dialogue system to support automatic enrollment of a person's first and last names in the ORION task delegation system [9]. This paper addresses our further goals towards a speak-and-spell solution in situations where the spoken and spelled utterances are disjoint. We envision that an unknown word would be uttered in the context of a carrier utterance, such as "What is the phone number of the *Thaiku* restaurant?" A subsequent spelling of "Thaiku" in a follow-up subdialogue would be processed jointly with the extracted pronounced word. Once the name is known, the phone number could be looked up via a Web search based on the name and presented to the user.

The remainder of this paper begins by describing advances in the NL server, which enable grammar updates, along with subsequent regeneration of the speech recognizer's search space. We then present refinements in the sound-to-letter technology, and experiments that assess the feasibility of artificially composing a "speak and spell" utterance to account for cases where the two resources are uttered separately. We conclude with a summary and a discussion of future work.

2. Augmenting the System with New Words

In a previous paper [2], we reported a novel algorithm to deduce the spelling and pronunciation of a "spoken and spelled" unknown word using sound-to-letter technology. This algorithm was applied to process new user names during the enrollment phase of the ORION dialogue system. This section describes the process that ensues once the system has determined and verified the spelling of the new words. We focus on enhancements to the NL server, as well as interactions among the NL and recognizer servers, which are mediated via a control program executed by a central hub, within the Galaxy Communicator architecture [11]. Other aspects of the hub program involved in the execution of the speak-and-spell recognition phase are detailed in [2].

The NL components of our dialogue systems employ the TINA framework [8], which utilizes a set of context-free rules to define allowable utterance patterns within each domain-dependent grammar, along with a feature unification mechanism to enforce agreement constraints and handle movement phenomena. A superimposed spacio-temporal probability model applied to the parse tree structure provides significant additional constraint for selecting the best candidate hypothesis from a word graph proposed by the recognizer. The probabilities are acquired by parsing a large training corpus and tabulating frequency counts on observed patterns.

[1] The research at MIT is supported by an industrial consortium supporting the MIT Oxygen Alliance and by DARPA under contract number NBCH1020002 monitored through the Dept. of the Interior, National Business Center, Acquisition Services Div., Fort Huachuca, AZ. The research at CNRI is supported by DARPA under contract number N66001-00-2-8922, monitored through SPAWAR Systems Center, San Diego.

Until now, the NL server typically preloads a trained grammar for each domain. The grammar probabilities are computed *a priori* during the training phase by observing up to several hundred thousand utterances. Hence, the trained models, by nature, are static throughout the lifetime of the running system. Previously, incrementing the grammar by a single word would necessitate the retraining of the entire grammar, followed by reloading. This would require several minutes to execute, making real-time operation infeasible. Current advances enable the adding of new words to the grammar *in real-time*, via several changes in the TINA framework. The resultant updates to the *trained* grammar is designed to be as close as possible to the trained grammar that would be created by completely retraining from a human-modified rules file.

Another important part of this new-word addition procedure lies in the adoption a new capability that automatically regenerates a class n-gram language model from the NL grammar [12]. The language model is ultimately converted to a finite-state transducer (FST) to be reloaded into the recognizer. Here too, the process involves parsing the corpus in order to tag selected words for their corresponding class assignments, a procedure that would require too much time to be practical in incremental updates. Hence we needed to not only incrementally update the context-free rules and the trained NL grammar, but also incrementally update the tagged corpus for the n-gram, along with its word-class assignments.

We will use the ORION user enrollment example to illustrate how the process works. Once the user has confirmed the spellings of both their first and last names, the hub sends to the NL server a frame detailing the spellings and pronunciations of both the first and last names, as determined by the letter-to-sound system, along with an identification of the assigned class (which is $user\_name$ in this case). The NL server then launches a procedure which completes the following steps:

1. compute an estimated count, N, of the likely number of occurrences of the new user name in the corpus, by averaging counts among all observed user names in the training corpus,
2. artificially augment the training corpus with N instances of the new user name, both in the raw untagged corpus and in the class-tagged corpus. This is important for future updates initiated by a system developer,
3. augment the NL vocabulary with any new words contained in the user's name and the grammar by adding the new user's name to the children of the category $user\_name$; updating the observation counts for this new entry to be N,
4. recompute the probability model for the $user\_name$ category from the artificially modified counts,
5. write the updated trained grammar to file, for subsequent system restarts; update control files for the recognizer's class n-gram,
6. launch an update of the recognizer's vocabulary and baseforms[1], as specified by the letter-to-sound system, rebuild the recognizer's class n-gram language model from the updated tagged corpus [12], and recreate the recognizer's fully composed FST, and
7. instruct the recognizer to reload its FST for this domain.

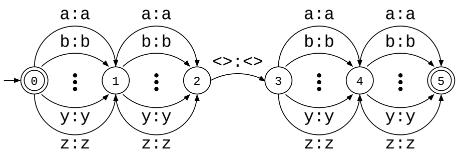

Figure 1: *Illustration of a length constraint FST L_i for $i = 2$. L_i is applied to enforce an equal number of letters proposed in the speak and spell parts of the waveform.*

While the current implementation completely rebuilds the recognizer FST with subsequent reload, a parallel effort [7] will in future bypass this step entirely, leading to an extremely efficient update of both the NL server and the recognizer server, such that the user would be able to speak a newly added word in the very next turn.

3. Sound-to-Letter Technology

In [2], we adopted an approach that uses a sequence of two recognition passes on the speak-and-spell waveform. The first stage is a simple letter recognizer augmented with an unknown word model to account for the preceding spoken word. It outputs an FST representing a graph of hypothesized letter sequences. In an intermediate stage, this graph is composed with an FST mapping pronunciations to letters, obtained through a sublexical framework ANGIE [10], trained on a large corpus of first and last names, as described in [1, 2]. A syllable-based n-gram language model provides additional constraint. An FST accounting for both the spoken and spelled portions is created by concatenating the sound-to-letter FST with the hypothesized letter graph. The resulting FST defines the search space for a final sound-to-letter recognition pass. The following section presents modifications to this initial approach for deriving improved results.

3.1. Imposing Further Constraints

In [2], hypotheses are obtained by simply selecting the highest scoring output of the pronounced portion of the final recognizer. Clearly, given the "speak and spell" mode, further constraint can be imposed on the system. As formulated in [2], the spoken and spelled portions are constrained by the same letter graph derived from the first pass. The fact that both solutions must represent the *same* spelling is ignored. But this constraint could easily be applied to an N-best list in a *post-processing* stage. However, we consider alternative ways in which this knowledge can be applied earlier, prior to the second sound-to-letter recognition.

We have investigated two strategies that are based on modifications to the FST input to the second-stage recognizer, using simple FST composition techniques. In the first method, a length constraint is imposed, such that the spoken and spelled hypotheses are required to contain the same *number* of letters. The second method enforces a much stronger constraint that the two hypotheses must be identical. Results are compared with a system which enforces consistency only after further analysis of a variable length N-best list.

3.1.1. Word Length Equality Restriction

Let F represent the output FST of the intermediate stage. As described in [2], F defines the compact search space from the first-stage letter recognition, and the application of various language constraints, including a statistical sound-to-letter model [10]. Mapping phones directly to letters, F is a concatenation of two

[1] If a prior pronunciation exists for a new word in the general lexicon, it is combined with the proposed pronunciation in a single baseform lexical entry.

FSTs representing the spoken and spelled part of the utterance:

$$F = F_1 \cdot F_2 \tag{1}$$

where F_1 incorporates sound-to-letter mappings for the spoken part and F_2 supports the spelling recognition.

The algorithm relies on F which mandates the output of a marker (<>) from the end of the spoken word to the beginning of the spelled part for all paths. Therefore, we can construct an FST L_i that licenses only paths for words of length i, in speak and spell mode. An example of L_i for $i = 2$ is illustrated in Figure 1. Then, FST H_i is created by the composition:

$$H_i = F \circ L_i \tag{2}$$

Essentially, H_i captures the portion of F's search space where the spoken and spelled hypotheses are all words of length i. The composition is performed for all i up to a maximum set arbitrarily to 20. Subsequently, the final FST is K is the union of all H_i's.

$$K = \bigcup_{i=1:max} H_i \tag{3}$$

K has extracted, from the original search space F, only those paths where the number of letters in the speak and spell part agree, up to a maximum length.

3.1.2. Word Equality Restriction

The above method can be augmented to restrict the letter candidates in the two parts to be identical. This is performed as an additional step in which FST compositions ensure that within the search space, if l_j is a candidate for the j^{th} letter in the spoken part, then the j^{th} letter in the spelling part must also be l_j. As in the length constraint method, the equality constraint relies on a composition with an FST M_{i,l_j}. M_{i,l_j} is similar to L_i (illustrated in Figure 1), except that the j^{th} letter is restricted to be l_j. The algorithm is outlined below.

```
for $i = 1..max$
  Compute $H_i = F \circ L_i$
  Set $T = H_i$
  for $j = 1..i$
    Find $\mathcal{L}_j$ the set of all letters at position $j$ in $T$
    foreach $l_j \in \mathcal{L}_j$
      Compute $T_{l_j} = T \circ M_{i,l_j}$
    end
    Compute new $T = \bigcup_{\forall l_j \in \mathcal{L}_j} T_{l_j}$
  end
  Set $H_i = T$
end
Compute $K = \bigcup_i H_i$
```

3.2. Processing Separate Spoken and Spelled Utterances

The overall algorithm above extends naturally to any application where a new word has been spoken and subsequently spelled. In particular, we have implemented the case where the spoken word is in an isolated utterance separate from the spelled part. In a way, this poses an easier problem, obviating the need for an unknown word model in the letter recognition stage, and reducing the chances for alignment errors for the beginning of the spelling part. As in the original formulation, the intermediate stage utilizes an FST of letter hypotheses to constrain the search space for sound-to-letter recognition. In contrast with Equation 1, only F_1 is required.

	Test Set A IV Words		Test Set B OOV Words	
System	*LER*	*WER*	*LER*	*WER*
I	8.3	25.7	14.3	48.9
II ($N = 10$)	7.4	24.0	11.7	41.1
II ($N = 50$)	6.8	23.3	11.7	41.1
III ($N = 1$)	6.9	21.9	12.0	43.8
III ($N = 10$)	6.8	23.3	11.7	41.1
IV ($N = 1$)	7.0	23.6	11.6	41.1

Table 1: *Letter Error Rates (LER) and Word Error Rates (WER) in percentage for two tests sets representing in-vocabulary IV (Test Set A) and OOV (Test Set B) results.*

With the above modified approach, imposing mutual restrictions on the spoken and spelled parts, a logical next step is to impose similar constraints for the separate spoken and spelled waveforms. A simple implementation is to concatenate the two waveforms together, and perform the sound-to-letter recognition stage on the concatenated waveform. Meanwhile, the input FST to this stage can be computed as described above, first concatenating two subcomponents as in Equation 1, and then imposing restrictions described in the previous sections.

4. Experiments

Evaluations have been conducted on several unseen test sets. A first set of experiments involves data containing an open set of spoken and spelled names of telephone-quality speech, the same as those used in [2]. The second set of experiments use data from the OGI Spelled and Spoken Word corpus [3], where the spoken and spelled names are recorded in separate utterances.

4.1. Speak and Spell Mode

As in [2], results are reported for a Test Set A, containing 416 words that have been previously observed in ANGIE's training vocabulary of 100,000 names, and a Test Set B, containing 219 words, that are considered out-of-vocabulary (OOV). System I is the baseline system which does not enforce any constraints between hypotheses from the spoken and spelled parts of the waveform. System II examines the N-best output in a post-processing stage, to seek agreement, if possible, in the spoken and spelled portions. We report results for $N = 10$ and $N = 50$. System III adopts the algorithm which enforces the same number of letters for the spoken and spelled parts as in Section 3.1.1. We report the results for the top-scoring hypothesis as well as using the additional post-processing stage on a 10-best list. Finally, System IV enforces agreement between the spoken and spelled portions in the intermediate stage as in Section 3.1.2. The composite results are tabulated in Table 1.

4.2. Separate Spoken and Spelled Utterances

In the next experiment, results are obtained for data that combines information on the spoken and spelled word, which are recorded in separate utterances. Our test set contains 2,388 surnames, where, for each name, the spoken and spelled versions are represented in separate isolated utterances. Of these, 1,967 names occurred in the ANGIE training data and 421 names are OOV. System V represents results obtained from the top-scoring output of the first-stage letter recognizer. System VI corresponds to the algorithm where the letter graph derived from the spelled utterance is used to define the search space to recognize

System	IV		OOV	
	LER	*WER*	*LER*	*WER*
V	10.7	38.1	16.7	60.3
VI	6.0	23.3	11.9	46.3
VIIa ($N = 1$)	5.5	20.5	11.7	44.4
VIIb ($N = 1$)	5.7	21.4	10.8	41.6

Table 2: *Letter Error Rates (LER) and Word Error Rates (WER) in percentage for IV and OOV in 2,388 names with spoken and spelled data taken from separate utterances from OGI test set.*

the spoken waveform. In System VII, the spelled and spoken waveforms are concatenated together, simulating a speak-and-spell mode, so that the algorithm described in [2] is employed and furthermore, we investigate alternatives that only license paths where (VIIa) the number of letters in the spelled and spoken part are equal, or (VIIb) exactly the same letters are proposed in the spelled and spoken parts. Results are shown in Table 2.

4.3. Discussion

For the speak-and-spell data, the post-processing approach using $N = 10$ alone has been able to achieve the optimal result for the OOV Test Set B, although for Test Set A, examining a deeper N-best list is better. This performance is matched by the system that enforces the same number of letters in the two portions of the waveform, and using a shallower N-best list of 10. That is, the word length restriction promotes better hypotheses towards the top of the list. Imposing the stronger constraint in System IV also yields significant improvements from the baseline System I, although performance does not exceed that of System III for Test Set A. The current ORION system has adopted the strategy used in System III, where the post-processing stage seeks equality between the spoken and spelled parts for an N-best list of 10.

The results on the OGI data allow us to measure performance gains obtained by integrating the sound-to-letter constraints into the letter recognition task. First, the improvements between System V and the remaining systems are a direct consequence of integrating the spoken name and the sound-to-letter model. Both methods in System VII that utilize the mutual constraints of the pronunciation and the spelling yielded further improvements compared with System VI. However, no further improvements from the post-processing stage (omitted from the table) are obtained. These overall results demonstrate that (1) using the spoken name can enhance the letter recognition task, and furthermore (2) in treating the task as a speak-and-spell mode, simultaneous constraints are successfully afforded on the two spoken and spelled waveforms, integrated into a single search.

5. Summary and Future Work

This paper has described our recent efforts towards the goal of empowering end users to augment the capabilities of spoken dialogue systems through natural spoken interaction. We have finally reached a milestone with the demonstrated ability to enroll a user's first and last names through interactive dialogue, by soliciting and processing a "speak-and-spell" spoken input.

In future work, a first step will be to incorporate the research described in [7] to enable incremental update of the recognizer's *preloaded* FST, which should reduce the time required to update the recognizer from a minute to a fraction of a second. This is critical for scenarios where the user may want to speak the new word in the very next utterance, and will enable the larger goal of updating the vocabulary from a list of named entities retrieved from the Web.

In the future, we envision that a dialogue system could reconfigure itself on the fly to support a *set* of proper nouns that are retrieved from a Web site in order to specialize its memory for a particular city. Thus, a user might send e-mail to ORION instructing it to call them "to discuss restaurants in Seattle." ORION would then immediately download a set of restaurant names from a Seattle-based Web site, and augment its restaurant class to support these names. The sound-to-letter component, working with error-free spellings[2] but without spoken utterances as examples, would generate multiple hypothesized pronunciations for the names, and populate the recognizer and NL components with the results.

In line with the above discussion, future experiments will involve general classes of unknown words such as names of geographical locations or businesses. A promising alternative to the "speak-and-spell" model is a model where the spoken form of the word is excised from the context of a prior utterance, and a spelling is separately solicited in a follow-up subdialogue. We have shown here that the speak-and-spell model can work well by concatenating the two waveform segments spoken in isolation; it remains to be seen how well the technology will perform on an automatically detected embedded unknown word.

6. References

[1] G. Chung and S. Seneff, "Integrating speech with keypad input for automatic entry of spelling and pronunciation of new words," *Proc. ICSLP '02*, Denver, CO, pp. 2061–2064, Sep. 2002.

[2] G. Chung, S. Seneff and C. Wang, "Automatic acquisition of names using speak and spell mode in spoken dialogue systems," *Proc. HLT-NAACL '03*, *to appear*, Edmonton, Canada, May, 2003.

[3] R. Cole et al., "A telephone speech database of spelled and spoken names," *Proc. ICSLP '92*, Banff, Canada, Oct. 1992.

[4] S. Dusan and J. Flanagan, "Adaptive dialog based upon multimodal language acquisition," *Proc. ICMI '02*, Pittsburg, PA, Oct. 2002.

[5] A. Gorin, "On automated language acquisition," *Journal of the Acoustical Society of Amrcia*, 97(6), pp. 3441-3461, 1995.

[6] D. Roy et al., "A trainable spoken language understanding system for visual object selection," *Proc. ICSLP '02*, Denver, CO, pp. 593–596, Sep. 2002.

[7] J. Schalkwyk, L. Hetherington, and E. Story, "Speech recognition with dynamic grammars using finite-state transducers," *submitted to Eurospeech '03*.

[8] S. Seneff, "TINA: A natural language system for spoken language applications," *Computational Linguistics*, Vol. 18, No. 1, pp. 61–86, 1992.

[9] S. Seneff, C. Chuu, and D. S. Cyphers, "ORION: From on-line interaction to off-line delegation," *Proc. ICSLP '00*, Vol. II, pp. 142–145, Beijing, China, Oct. 2000.

[10] S. Seneff, R. Lau, and H. Meng, "ANGIE: A new framework for speech analysis based on morpho-phonological modelling," *Proc. ICSLP '96*, vol. 1, pp. 110–113, Philadelphia, PA, 1996.

[11] S. Seneff, R. Lau, and J. Polifroni, "Organization, communication, and control in the GALAXY-II conversational system," *Proc. Eurospeech '99*, pp. 1271–1274, Budapest, Hungary, Sep. 1999,

[12] S. Seneff, C. Wang, and T. J. Hazen, "Automatic induction of n-gram language models from a natural language grammar," *submitted to Eurospeech'03*.

[2] Although abbreviations would become a new issue.