AUTOMATICALLY INCORPORATING UNKNOWN WORDS IN JUPITER[1]

Grace Chung

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 USA
http://www.sls.lcs.mit.edu, mailto:{graceyc, seneff}@mit.edu

ABSTRACT

This paper concerns the handling of out-of-vocabulary (OOV) words in the JUPITER weather information system. Specifically our objective is to deal with weather queries regarding unknown cities. We have implemented a system which can detect the presence of an unknown city name, and immediately propose a plausible spelling for that city. Potentially, the city can be dynamically incorporated into the recognizer lexicon. The three-stage system described in [1] was implemented in the JUPITER domain, and this paper will detail the development of a system that uses an ANGIE-based framework to model both spelling and pronunciation simultaneously, and uses automatically derived novel lexical units in the first stage. We report results on an independent test set containing unknown cities. Compared with a single-stage baseline, word error was reduced by 29.3% (from 24.6% to 17.4%) and understanding error was reduced by 67.5% (from 67.0% to 21.8%) on the three-stage configuration.

1. INTRODUCTION

For most conversational systems today, the gap in performance between sentences with OOV words and in-vocabulary sentences remains wide. It is important for systems to detect the incidence of unknown words and handle them intelligently. In the JUPITER domain, there frequently arise sentences containing unknown words. It was reported in [5] that, while word error rate for in-vocabulary test sentences was at 8%, the error for test sentences with unknowns escalated to around 50%. These sentences may consist of queries that are entirely beyond the scope of the domain; they may be sentences contaminated by word fragments and other artifacts of spontaneous speech; or often, they are within-domain queries that stretch beyond the limits of system knowledge, for instance, weather information for unknown cities. In all the above cases, the system may reject the sentence or commit multiple recognition errors by hypothesizing acoustically-similar words in place of the OOV item, further confounding the dialog. Because no feedback is offered to the user regarding the source of the problem, there exists no opportunity for error recovery. Our work specifically addresses sentences where it might be possible for the system to identify the unknown parameter in the query; that is, for JUPITER, we envision the system to inform the user whenever weather for a city in question is unavailable. Furthermore, by being able to detect the presence of an OOV item, we hope to improve overall recognition accuracy for the sentence, narrowing the divide in performance between in-vocabulary and OOV sentences. As the system makes phonetic and spelling hypotheses for the unknown word, it is also possible to dynamically grow the lexicon by this previously unseen city name.

As a first step towards this goal, this paper describes the course of implementing the flexible vocabulary system discussed in [1] in the JUPITER domain. In [1], we conceived of a three-stage solution for the general problem of dealing with open and dynamically extensible vocabularies. Here, we test the feasibility of our ideas on sentences with unknown city names. The system is tailored to automatically detect unknown cities, and propose phonetic and orthographic transcriptions for them such that they could be incorporated into the lexicon on the fly.

The next sections will examine various aspects of our JUPITER -based system, and also present performance results obtained on test utterances containing unknown cities. Section 2 outlines the three-stage architecture. Incorporated in the first stage are ANGIE [4] probabilistic models where pronunciation and spelling could be modeled simultaneously via a spelling grammar. The details and utility of special *letter-phoneme* units, embedded in the grammar, are described in Section 3. The first stage uses lexical units that are derived by a data-driven, iterative method, conceived in [1]. In Section 4, we discuss the outcome of this procedure and the nature of the new word and morph units, and explore the ramifications of modeling low-level sublexical information using an entirely novel set of automatically generated lexical units. The final sections will describe experiments undertaken with sentences containing unknown cities.

2. SYSTEM ARCHITECTURE

As detailed in [1], the current system consists of three stages. In the first stage, we pre-load a single finite-state transducer (FST) expressing all language constraints. This FST is pre-composed from the following: $C \circ P \circ L \circ G$. C transduces context-dependent labels to context-independent phone units. P is an ANGIE-derived column-bigram FST which maps phones to letter-phonemes (described in Section 3). The lexical units of this recognizer are computed in two steps. Initially the iterative procedure first outlined in [1], is implemented, resulting in novel morph units (presented in Section 4). Secondly, for the final lexicon, we assemble a set of morphs and sub-morphs drawn from the novel units, that is, a set of onsets and rhymes, decomposed from the stressed morphs, plus the set of unstressed morphs. L maps the letter-phonemes to these lexical units, and G represents the trigram information on these lexical units.

Our main focus is to achieve high recognition accuracy on in-vocabulary while providing linguistic support to novel sequences for unknown words. And the solution to this is maximizing low-

[1] This material is based upon work supported by the National Science Foundation under Grant No. IRI-9618731.

level linguistic constraint in the first stage by combining various knowledge sources that are generic to all English vocabulary. Conventional sources of low-level information are derived from the phone or syllable. In our case, as the first stage reduces its morph hypotheses back into a phonetic lattice, we are not tied to a fixed set of lexical units, and are free to re-organize the lexical space into a more efficient one. This is done by our iterative procedure whose outputs are novel word and morph lexicons. As will be seen, the resultant words resemble metrical foot units, reflecting the stress and rhythmic patterns of a sentence. We ultimately train our ANGIE grammar on the new lexicons, and generate a column-bigram FST. Consequently, the word substructure patterns captured by ANGIE are derived not from the original vocabulary but from the automatically generated footlike units. Furthermore, the trigram information is also based on them.

The second-stage search is guided by a phonetic lattice, output from the first. Here, the ANGIE parse mechanism is combined with a word bigram to constrain phonetic hypotheses. Unknown word hypotheses are only permissible at restricted locations. If an ANGIE grammar with letter-phonemes is used in stage two, spellings of unknown words can be hypothesized instantaneously. At this point, an N-best list is produced, and subsequently converted to a word network. Finally, our natural language module (NL), TINA [3], parses the word graph to output the highest-scoring sentence hypothesis and a meaning representation.

3. LETTER-PHONEMES FOR JUPITER

As argued in [1], probability models that capture grapheme information in conjunction with phonological phenomena may lead to enhanced linguistic constraint while providing the convenience of sound-to-letter capability within the recognition framework. In the ANGIE grammar, this would involve constructing a set of units that resides at the pre-terminal layer of an ANGIE parse tree, and codifies spelling and phonemic information at the same time. We refer to these as *letter-phonemes*, designed by selecting a set of grapheme units, and augmenting them with carefully chosen characteristics that distinguish phonemic correspondence and linguistic context. Similar to the original grammar, a set of hand-written context-free rules specifies the allowable phonetic realizations of each letter-phoneme. Effectively, these letter-phonemes are subdividing the phoneme space into more specific units, resulting in finer-grained probability modeling, thereby affording tighter constraint. The resulting parse tree characterizes generic word substructures, phonological processes as well as spelling rules.

As the lexicon in ANGIE is organized into two tiers, vocabulary words are defined in terms of their morph baseforms whereas morphs are defined by their phonemic sequences. Therefore, for the new grammar, each morph is associated with a letter-phoneme sequence from which both the morph spelling and pronunciation can be inferred. During recognition, upon encountering novel letter-phoneme sequences, a potential spelling can be deduced instantaneously by concatenating the proposed letter-phonemes stripped of their peripheral markers. For example, the sequence *p! l a_l+ te* [2] can be concatenated together to form the word, "plate" (which consists of a single stressed morph).

The letter-phoneme categories are chosen by hand in an attempt to reduce perplexity and improve predictive performance in both letter-to-sound and pronunciation variation. Following are some characteristics we have included:

- *Vowels:* These are marked for stress by "+," and for distinctions between long and tense vowels. For example, *i_l+* is the letter "i," in "like," which is phonetically realized as a long stressed vowel, that is, it maps to /ay/ or /iy/ in a stressed syllable. *a_x+* is the letter "a," in "add," which is realized as a tense stressed vowel, that is, it maps to /ae/, in a stressed syllable. As with the original grammar, some vowels within function words are modeled separately, e.g. *ee_fcn* for the vowel within the function word "been."
- *Consonants:* When in the syllable-onset position, consonants are marked with "!." Some letter-phonemes have multiple phonemic correlates such as the coda *"gh"* which can be realized as /f/ or /g/. Multiple letter-phonemes may have the same phonetic realization. For example, *n* and *ne* are different spellings for /n/ in coda position. The probability models will learn that long vowels are more likely to precede *ne* than *n*. Other examples are *ti!* as the onset in the *tion* suffix, realized by /sh/ and letter-phonemes which capture diphone contexts such as *nch*, *nd* and *nk*.
- In the instances where training data are sparse, some graphemes are collapsed together into a single letter-phoneme, forming a more generic model. In doing this, some spelling information is discarded and cannot be recovered. For example *ain* (in "mountain") and *oln* (in "Lincoln") are spelling variations of an unstressed rhyme realized as /en/. They are merged together into one model due to insufficient data. This amounts to a trade-off in foregoing some sound-to-letter capability.

There are in total 289 letter-phoneme categories compared with 115 phonemes in the original JUPITER grammar. The grammar is trained from about 50,000 JUPITER sentences that were forced-aligned from a baseline SUMMIT recognizer [5]. On a test set of 425 utterances, per phone perplexity, computed using the original phoneme grammar and the new spelling grammar, is found to improve from 5.7 to 5.3, respectively. This reduction in perplexity may directly benefit recognition performance.

4. NOVEL LEXICAL UNITS

An iterative procedure is employed to construct a novel set of lexical units for the first-stage lexicon. This procedure involves repeatedly constructing an ANGIE FST from the training data, and then searching for the best scoring path in order to seek letter-phoneme sequences that better model the given phonetic realization of the training data. The process first uses an ANGIE FST constructed from the initial spelling grammar. At each iteration, for every individual sentence, a likely letter-phoneme sequence and the morph class identity (e.g. stressed root, suffix, prefix and so on) are output. New morphs and words are created by concatenating the novel letter-phoneme sequences, forming the inputs to the new grammar. At the last iteration, the new lexicons are used to train an ANGIE grammar from which a column-bigram FST is constructed. During recognition, the first stage is subjected to only hypothesize lexical units derived from the novel morph lexicon, and does so with the aid of ANGIE's implicit sublexical knowledge and trigram models also trained from the novel morphs. The new units are characterized by novel spellings because they are computed by concatenating novel letter-phoneme

[2] The meaning of the markers will be elucidated later.

sequences. In fact, as these pseudo-words are specified to contain exactly one stressed morph with optional prefixes and suffixes, the new lexical organization seems to capture the stress and rhythmic patterns in the acoustic realization of the sentence. For more details of the procedure, refer to [1].

In implementing the algorithm, it was found that after four iterations, the ANGIE grammar converges, and we arrive at our final lexical units. On our 425-utterance test set, per phone perplexity for the grammar at each iteration steadily falls, and it settles at 4.9 in the final iteration, a 10% reduction from the original grammar. Using the final grammar, a column-bigram FST is generated. Table 1 below compares the column-bigram FSTs constructed from different grammars. The original spelling grammar requires a much larger FST than a grammar using phonemes only. But the final iterated grammar achieves a smaller FST using letter-phonemes in its models. Several letter-phonemes are never chosen by the algorithm, and they are discarded. Thus the letter-phoneme set is reduced to 264 from 289. By contrast, the size of the morph lexicon increased from 1927 to 2071, and the word lexicon size increased from 2011 to 3516. But with the stressed morphs decomposed to onsets and rhymes, the recognizer lexicon totals only 900 morph and sub-morph units. The final FST composed with the trigram model is minimized and fully determinized. It has around 4 million arcs and 440k states, occupying 90Mbytes of memory, which is less than half the FST size of the original phoneme grammar.

Grammar	Arcs	States
Original Phoneme	9385	1488
Letter-phoneme	12k	2175
Final	9741	1717

Table 1: Size of FSTs with Different ANGIE Grammars.

We proceed to examine more closely the nature of the novel units. A large portion of the morphs and words remained unchanged during the algorithm. Characteristics of the new units that emerged are documented below.

- Some words and morphs have changed in spelling because the algorithm preferred an alternative letter-phoneme sequence to the original designated one, e.g., *lundon* for London, *edmanton* for Edmonton, *kuwate* for Kuwait and *mareen* for marine. This signifies that the alternate letter-phoneme yields a higher probability, thereby reducing perplexity, and it directly leads to a reduction in the number of letter-phoneme categories in the set.

- Because the algorithm chooses the best letter-phoneme sequence for the phonetic realizations of a sentence without regard to original word boundaries, at times, letter-phonemes that reside at word boundaries switch word affiliation, creating novel words. For example, "July ninth in .." is changed to "*juline ine thin* ..". Here there are two instances of consonants at word boundaries changing between onset and coda position. In the first case, where the phone sequence begins with /jh uh l ay n/, improved probabilities are attained from the morphs *ju- line+ ine+* than *ju- ly+ nine+*[3]. Similarly, /th/ at the end of nine is originally modeled as an inflexional suffix but is found to be more beneficial as a syllable onset.

- It is found that the steady increase in the word lexicon size is caused by many novel words being created. Some adjacent words are clustered into a single one where one syllable is assigned lexical stress and the remainder are identified as prefixes or suffixes, e.g., the single words *a- good+ -day*[4] for ".. a good day" and *to- rain+* for ".. to rain .." This seems to characterize the alternating rhythmic pattern of the sentence. Another related phenomenon is where, in some word pairs, the suffix of the preceding word changes to the prefix of the next or vice versa. Some examples (with the spelling changes included) are the words *sandi ego* for San Diego and *atlan togeorgia* for Atlanta Georgia.

The success of this algorithm in reducing both FST size and perplexity suggests that our novel units could provide enhanced constraint and efficiency for the first stage. The recognition experiments will establish whether these units will afford sufficient coverage for sequences within previously unseen words.

5. UNKNOWN CITY EXPERIMENTS

5.1. Stage Two and Three

During the second stage, generally, unknown words may be hypothesized if their phonetic sequences can be admitted by the dynamic ANGIE parse mechanism. The parser employs a JUPITER-based grammar with the original phoneme set at the pre-terminal layer. An empirically-determined unknown word penalty is added to the word score, and to restrict the length of an unknown word, it may only contain one stressed morph. In addition, as only the city name category is expected for an unknown in our test utterances, we impose the constraint that unknowns are permitted to occur exclusively following a short list of words. These are found to be words preceding city names in the training data. They are provided in the following: *in, for, uh, oh, um, on, at, is, about, what_about, like.*

In order to train TINA to handle unknowns appropriately, we employed a heuristic approach. It is chosen randomly that for one out of every ten training sentences containing a city name, the city name is replaced by an "unknown" tag. During training, as TINA encounters the tag in a sentence, the tag is treated as an unknown city category, thereby boosting the probabilities for unknown city names. Within the 56k training sentences, there were approximately 2000 sentences that were artificially augmented with the "unknown" marker. This is merely a simplified approach and can be replaced by a more sophisticated training technique in future.

5.2. Experimental Detail

In our experiments, we evaluate performance on an independent test set of 425 utterances. This set has been chosen such that all sentences pertain to weather information queries regarding unknown cities. Therefore, each test utterance contains exactly one unknown city name. For comparison, our baseline performance is obtained from a single-stage SUMMIT [5] recognizer which does not have capability to handle OOV items. It uses the same context-dependent acoustic models as the three-stage system, and a bigram and trigram word model.

[3] The "-" appended to a morph marks the prefix.

[4] The "-" beginning a morph marks a suffix.

System	WER(%)	UER(%)
Baseline system	24.6	67.0
1: Two stage	15.6	31.3
2: Two stage with TINA	17.2	24.3
3: Three stage	17.4	21.8

Table 2: Word (WER) and understanding (UER) error rates for baseline system and three experimental systems on a 425-utterance test set with unknown city names.

For comparison, experiments are conducted on three variations of our system. We consider (1) a two-stage only version where the top scoring sentence hypothesis of the second stage is evaluated, (2) a two-stage version which employs TINA in stage two, and (3) the full three-stage system. In the three-stage system (3), the third-stage word network is constructed from the N-best list of stage two with $N = 20$. And the second stage in (2) integrates TINA with ANGIE in a control strategy described earlier in [2]. As described there, the motivation for using TINA in stage two is an attempt to utilize NL constraints earlier within the recognizer search. In this configuration, an unknown word is only hypothesized by stage two if an unknown city name can occur, according to the syntax and semantics stipulated by TINA. Results are reported for word (WER) and understanding (UER) error rate where the understanding evaluation measure is devised previously in [2], based on comparing key-value pairs extracted from a TINA-based meaning representation. A sentence is recognized/understood correctly if all the known words are recognized/understood, and an unknown flag is proposed at the place of the unknown city name.

In another pilot experiment, we test the feasibility of instantaneously proposing letter spellings for new words within the second stage. Instead of the original JUPITER grammar, we use the spelling grammar in the parse mechanism, which enables the extraction of spelling hypotheses directly from the phonetic hypotheses. At this preliminary stage, for the purpose of simplicity, TINA is omitted, and we report results for the top scoring hypothesis of the second stage.

5.3. Results and Discussion

WER and SER for the baseline and the three experimental systems are summarized in Table 2. The baseline system achieved a WER of 24.6% and UER of 67.0%. Upon closer examination, in spite of the incidence of exactly one unknown city per utterance, the system committed on average 1.9 errors per utterance. Significant improvements are made using System 1 with WER 15.6% (36.6% improvement) and UER of 31.3% (53.3% improvement). There are on average 1.2 errors committed per utterance. The detection error for unknown words is 21.% (2.6% false alarms and 18.6% misses.) These positive results were accomplished before the introduction of NL constraints, and establish the benefits of the system's ability to handle unknown words. For System 2, employing TINA directly within the recognition search, as well as allowing ANGIE to propose unknown words, has proven to be an expensive overhead, since computation time escalates. With the results of WER 17.2% and UER 24.3%, improvement is attained above System 1 but falls short of System 3. With the full three-stage system in use, WER is 17.4% (29.3% improvement from baseline) and UER is 21.8% (67.5% improvement from baseline). This indicates that the small word network interfacing a third stage is more effective. As consistent with trends recorded in [2], exploiting NL constraints improves understanding with a trade-off in word accuracy. In using TINA, many more sentences can be parsed for NL than in System 1, contributing to the superior understanding accuracy. After all, optimizing on understanding performance is the goal.

The above systems have produced very encouraging performance improvements. These can be attributed to the ability to maintain high in-vocabulary accuracy in concert with success in detecting unknown word occurrences. All this is accomplished at near-real-time speeds comparable with the baseline configuration.

In the pilot experiment, when the JUPITER grammar is replaced by the spelling grammar in stage two, the system performs overall at 16.5% WER and 32.5% UER. For the subset of sentences with 100% recognition accuracy, we computed the letter error rate of the 164 proposed unknown words. We found that the system achieves 57.8% error. Although this result remains preliminary, it nonetheless demonstrates that it is possible to extract spellings of unknown words during recognition time. We have ascertained these results without optimizing on our sound-to-letter capabilities. The second stage here requires marginally more computation time because of the increased size of the letter-phoneme grammar.

6. CONCLUSIONS

The above has presented a solution for automatically incorporating unknown cities names in the JUPITER domain. Our stategy includes a novel inital stage that employs low-level constraints, and uses an automatically generated lexicon whose units capture rhythmic characteristics of the sentences. These units are built from novel phoneme-level units encoding both spelling and pronunciation. Our experiments yielded promising results for the three-stage design. The final three-stage system exhibited the ability to detect OOV items among in-vocabulary words, to process them as unknown cities in the NL component, and to extract spelling hypotheses for the new cities.

7. ACKNOWLEDGEMENT

I would like to thank Dr. Stephanie Seneff for her supervision on this work, and her valuable comments on this paper.

8. REFERENCES

1. G. Chung, "A Three-Stage Solution for Flexible Vocabulary Speech Understanding," in these Proceedings.

2. G. Chung and S. Seneff, "Towards Multi-Domain Speech Understanding Using a Two-Stage Recognizer,", in *Proc. Eurospeech '99*, Budapest, Hungary, pp. 2655-2658, Sep. 1999.

3. S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," in *Computational Linguistics*, 18(1), pp. 61-86, 1992.

4. S. Seneff et al., "ANGIE: A New Framework for Speech Analysis Based on Morph-Phonological Modeling," in *Proc. ICSLP '96*, Philadelphia, PA, Oct. 1996.

5. V. Zue et al., "JUPITER: A Teleophone-Based Conversational Interface for Weather Information," in *IEEE Trans. Speech and Audio Processing*, 8(1), pp 85–96, 2000.