AUTOMATIC LEXICAL PRONUNCIATIONS GENERATION AND UPDATE

Ghinwa F. Choueiter, Stephanie Seneff, and James R. Glass

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA 02139, USA
{ghinwa, seneff, jrg}@csail.mit.edu*

ABSTRACT

Most automatic speech recognizers use a dictionary that maps words to one or more canonical pronunciations. Such entries are typically hand-written by lexical experts. In this research, we investigate a new approach for automatically generating lexical pronunciations using a linguistically motivated subword model, and refining the pronunciations with spoken examples. The approach is evaluated on an isolated word recognition task with a 2k lexicon of restaurant and street names. A letter-to-sound model is first used to generate seed baseforms for the lexicon. Then spoken utterances of words in the lexicon are presented to a subword recognizer and the top hypotheses are used to update the lexical baseforms. The spelling of each word is also used to constrain the subword search space and generate spelling-constrained baseforms. The results obtained are quite encouraging and indicate that our approach can be successfully used to learn valid pronunciations of new words.

Index Terms— Letter-to-sound model, lexical pronunciations

1. INTRODUCTION

Most automatic speech recognizers (ASR) use a dictionary that maps words to one or more canonical pronunciations. Such entries, also known as lexical baseforms, are typically hand-written by lexical experts. When ASR systems are deployed in applications that constantly evolve such as broadcast news transcription, music queries, or restaurant reservation systems, they require constant changes to their dictionaries to account for the addition of new words that are often application-specific keywords. One solution to this problem is to provide these applications with access to larger dictionaries, but this is not always advantageous. For example, in this research, we consider a 2k lexicon of valid restaurant and street names collected for a restaurant reservation domain. Examples of these words are *aceituna, jonquilles, lastorias, pepperoncinis, chungs*. Of these 2k words, 500 are found in a 150k dictionary, 600 words are found in a 300k Google subset, and 1.4k words are found in a 2.5 million Google subset [1]. Thus, even as the vocabulary size is dramatically increased, a substantial portion of the lexicon remains unfound. This is to be expected since the restaurant business is constantly in flux and new restaurants are always emerging. An alternative solution is to routinely and manually update the dictionary. However, this can be time-consuming and prone to error, particularly when the words are unfamiliar or foreign-sounding such as proper names or restaurants.

In this paper, we propose to automatically learn and update the phonetic baseforms of a lexicon using a letter-to-sound (L2S) model as well as spoken instances of words in the lexicon. To assess our approach, the generated baseforms are evaluated on an isolated word recognition task.

The task of automatically generating word pronunciations is not recent, and there has been some research in this domain using decision trees [2] and phonetic decoding [3, 4, 5, 6]. Several researchers have also addressed the problem of L2S modeling [7, 8, 9, 10]. This work is different in that it uses a linguistically motivated, context-free grammar (CFG) approach to develop a robust bi-directional L2S model [11] that is used to learn the *seed baseforms* of a lexicon. The seed baseforms are then updated by presenting spoken utterances of words in the lexicon to a subword recognizer and using the top N hypotheses as baseforms. This research is inspired by the work of Chung et al. [12] but differs from it in several respects. Whereas Chung et al. used a bottom-up subword framework, our system uses a top-down probabilistic parser that encodes pronunciation in the pre-terminal units, and encodes all the spelling variants in the terminals. We also implement and evaluate our model on a larger lexicon of 2k restaurant names. Finally, our model has a simpler notation scheme which ties directly to a phoneme notation typically used in phoneme-based dictionaries.

In this paper we address the following questions: (1) How well does the L2S model perform at automatically generating lexical baseforms? (2) How much improvement is obtained by generating baseforms using the spoken utterances and the subword recognizer? (3) How much improvement is obtained if the spelling of a word is used to constrain the search space of the subword recognizer?

*This research was funded by the Industrial Technology Research Institute.

Although we implement and evaluate our model on an isolated word recognition task, we envision our approach implemented in open-ended continuous-speech applications. For example, given audio waveforms and their corresponding word transcription, the CFG-based L2S and subword models can be used to automatically update the dictionary corresponding to the data. Other applications are open-ended spoken queries that allow users to introduce manual corrections in case of transcription errors. Both spoken utterances and corrections can be used to update the lexical baseform of a pre-existing word or add the baseform of a new word to the dictionary.

In the rest of this paper, the approach is described in Section 2. This includes the CFG-based subword model, the L2S model, and the subword recognizer. The data collection is described in Section 3, and the experiments in Section 4. The paper concludes with a discussion and summary in Section 5.

2. THE APPROACH

2.1. A CFG-based Bidirectional Subword Model

The context-free grammar (CFG) used in this research has been designed to encode positional and phonological constraints in sub-syllabic structures [11]. The decision to represent sub-syllabic as opposed to whole-syllabic structures is motivated by the hypothesis that the former would generalize better to unseen data. Syllables are primarily decomposed into onset and rhyme which become the subword pronunciation units that are used to generate lexical baseforms. Hence, the subwords are intermediate units between phonemes and syllables, and they only contain pronunciation information. The grammar also makes use of sonority rules within a syllable combined with maximal stress and onset principles to make informed decisions about syllable boundary locations. Apart from onset and rhyme, the following linguistically motivated categories are introduced in the CFG:

ambi which denotes *ambisyllabic*, is introduced for a subset of intersyllabic consonants to allow ambiguity in the syllable assignment.

affix accounts for the typically coronal consonants that violate sonority rules in the coda (e.g. crep*t*, six*th*).

first stressed/unstressed are introduced to capture the statistics of the first stressed and unstressed syllables by discriminating between them and the rest of the stressed and unstressed syllables in a word.

usyl which stands for *unstressed syllable*, denotes a set of combined onsets and rhymes that form frequently occurring unstressed syllables.

The CFG uses 13 rules to describe the sub-syllabic structure of words. The following is a sample of those rules:

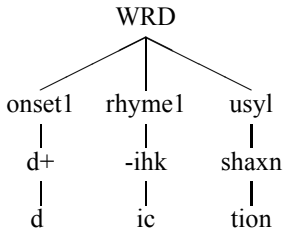

Fig. 1. Parse tree of the word *diction* obtained using the CFG.

```
WRD ⟹ rhyme1 (affix | usyl)
WRD ⟹ rhyme1 affix onset rhyme2 affix [affix]
WRD ⟹ onset1 rhyme1 [usyl] rhyme2 (usyl | affix)
```

where `[]` denotes optional and `(|)` denotes OR.

Further rules describe all possible ways the sub-syllabic structures map to subword units as well as all possible ways a subword unit can be spelled. Figure 1 illustrates in detail the parsing of the word *diction* into subword units such as onset and rhyme (denoted with + and - respectively). As can be seen from the parse tree, subword units are the preterminals and letter clusters are the terminals. The total number of pre-terminals and terminals are 677 and 1573 respectively. Furthermore, one by-product of the CFG is an automatically derived mapping between subwords and their spellings (preterminals and terminals), which results in hybrid units that contain both pronunciation and spelling information. The total number of these hybrid units, which we denote as spellnemes, is 2541. We illustrate below the parsing of a sample lexicon in terms of subword units.

```
abatements  -ax+ b+ -eyt maxnt +s
biderman    b+ -ih df -er maxn
demolition  d+ -ehm -axl -ih shaxn
```

Next, we illustrate the parsing of the same lexicon in terms of spellnemes. The `<\d+>` tag denotes the index to the pronunciation of the corresponding letter cluster. The pronunciation-to-spelling mappings can be used to create statistical bi-directional L2S models.

```
abatements  a<217> b<565> ate<378> ment<610> s<6>
biderman    b<565> i<385> d<573> er<332> man<608>
demolition  d<570> em<298> ol<226> i<385> tion<643>
```

A separately supplied lexicon maps the subword units to their phonemic realization as shown below:

```
-ayth   ay th
-ehb    eh bd
-uhng   uh ng
```

The framework underlying the CFG is a probabilistic top-down parser [13]. In this framework, the linguistic knowledge, which is manually encoded in the CFG, is complemented with statistics by superimposing the parse tree with a trigram model that captures the statistics of a child conditioned on its parent and its left sibling. We note that the main purpose of parsing is to align the letters with their pronunciations in order to automatically generate the spellnemes of the words.

2.2. The Letter-to-Sound Model

One of the advantages of the CFG-based framework is the ability to leverage from its tools to build an L2S model fairly easily. The task is facilitated by the spellneme units which encode both spelling and pronunciation information. The L2S model, T_{L2U}, is modeled using finite state transducers (FST) as follows [14]:

$$T_{L2U} = T_{L2SP} \, o \, G_{SP} \, o \, T_{SP2U} \tag{1}$$

where T_{L2SP} and T_{SP2U} are deterministic mappings from letters to spellnemes and from spellnemes to subwords respectively, and G_{SP} is a spellneme trigram. A search through T_{L2U} produces an *N*-best list of pronunciations corresponding to the input spelling. Hence, the L2S model is used to generate seed lexical baseforms automatically from the spellings.

2.3. The Subword Recognizer

The subword recognizer is used to automatically generate lexical baseforms from spoken utterances of words. The subword search space is implemented as a weighted FST, R:

$$R = C \, o \, P \, o \, L \, o \, G \tag{2}$$

where C denotes the mapping from context-dependent model labels to context-independent phone labels, P the phonological rules that map phone labels to phoneme sequences, L the subword lexicon, which is a mapping from phonemic units to subwords obtained from the CFG, and G the subword trigram. A search through R produces an *N*-best list of pronunciations corresponding to the spoken word.

In some of the experiments in Section 4.2, the subword search space is constrained with the spelling of the corresponding word in the lexicon. The L2S model, T_{L2U} is used to generate a constraining subword lattice, K, from the spelling of each word. The constrained subword search space, R_K is:

$$R_K = C \, o \, P \, o \, L \, o \, K \, o \, G \tag{3}$$

3. DATA COLLECTION

For the purpose of this research, a list of $\sim$2k restaurant and street names in Massachusetts is selected as the lexicon. These particular words are of interest to us because they form critical vocabulary in our multimodel restaurant guide domain [15]. The names are purposefully chosen to have relatively low Google hit counts. In order to automatically generate baseforms from data as well as to evaluate the approach, spoken instances of the words in the lexicon are required.

An online user interface was designed for the purpose of data collection. Each subject is presented with a word and is prompted to speak it. A subword recognizer complemented with a sound-to-letter model is used to generate hypothesized spellings of the spoken word. The spellings are then filtered using the 2k lexicon, and the top 5 candidates are presented to the subject. The framework used to hypothesize spellings from spoken utterances is similar to that in [16]. If the correct spelling is not in the proposed list, the subject is prompted to speak the word again. The same process is then repeated, and a new list of top 5 candidates is presented to the subject. If, again, the correct spelling is not in the proposed list, the subject spells the word. Although, in the future, we plan to integrate the data collected in spelling mode into our approach, it is not currently used in this research.

Excluding the data recorded in spelling mode, 2842 utterances were collected from 19 speakers, 12 males and 7 females. We note that each word is spoken by only one speaker. A breakdown and description of the collected data is shown in Table 1. As implied by Table 1, the lexicon of the First and Second set is one and the same.

Name	Size	Description
Single	1142	Words that were spoken once
First	850	First utterance of words spoken twice
Second	850	Second utterance of words spoken twice

Table 1. Description of the collected data. A total of 2842 utterances are obtained for a 2k lexicon.

4. EXPERIMENTS AND RESULTS

In all our experiments, the SUMMIT segment-based speech recognition system is used [17]. Context-dependent diphone acoustic models are used and their feature representation is based on 14 MFCCs (Mel-Frequency Cepstral Coefficients) averaged over 8 regions at hypothesized phonetic boundaries. The diphones are modeled with diagonal Gaussian mixture models with a maximum of 75 Gaussians per model, and are trained on telephone speech. The spellneme trigram, G_{SP}, is built with 55k parsed nouns extracted from the LDC pronlex dictionary. The subword trigram, G, is generated from 300k Google words parsed with the L2S model. Finally, the isolated word recognizer has a 2k vocabulary as described in Section 3, and only uses a word unigram.

Section 4.1 describes the automatic generation of the phonetic baseforms using the L2S model and reports on results. Section 4.2 describes the baseform update process which uses spoken instances of the lexicon, and the subword recognizer.

Results are reported for baseforms generated with the unconstrained as well as spelling-constrained subword recognizer. Section 4.3 reports the results obtained when the baseforms generated by the different setups are combined.

4.1. Pronunciations generated with the L2S model

In this section, we report the results obtained for the phonetic pronunciations automatically generated with the L2S model described in Section 2.2. First, the 2k lexicon is presented to the L2S model and the top$_n \mid n = 1, ..5, 10, 20, 50$ *seed* baseforms are generated for each word. We illustrate below the top 2 L2S seed baseforms for two sample words:

```
yainnis : ( y ay n ax s | y ey n ax s )
shawarma: ( sh ao aa r m ax | sh ax w aa r m ax )
```

We evaluate the baseforms on the 2842 utterances and report the results for the three sets, Single, First, and Second in Table 2. We first observe that the Single set has a lower word error rate (WER) than the sets First and Second. This is to be expected since Single is the set of words that is recognized in the first round during data collection and is likely therefore to be an *easier* set than First and Second. Next, the WER of Second is lower than that of First. One possible explanation is that subjects tend to speak the words more carefully on the second round upon failing the first one. Finally, as expected, the WER improves significantly as the number of alternative baseforms is initially increased, however the WER starts deteriorating as pronunciation confusion is increased, in this case beyond 20 baseforms.

	Single	First	Second
top1	25.7	52.4	47.8
top2	20.3	47.9	42.8
top3	17.9	47.6	41.2
top4	17.3	47.3	39.9
top5	17.1	47.8	39.5
top10	16.5	47.5	40.0
top20	16.9	48.5	40.2
top50	18.6	47.8	42.6

Table 2. WER of the three data sets, Single, First, Second as a function of the top$_n \mid n = 1, ..., 5, 10, 20, 50$ baseforms generated by the L2S model.

For comparison purposes and to evaluate the effectiveness of the L2S model at generating the lexical baseforms, manual corrections are carefully introduced into the top1 baseforms obtained with the L2S model. As shown in Table 3, absolute improvements of 2.2%, 1.9%, and 3.1% are obtained for the Single, First, and Second sets respectively. The modest improvement obtained following manual corrections is encouraging since it indicates that the L2S model is very good at generating valid pronunciations. In fact, in comparing Table 2 with Table 3, it can be noted that just 2 automatically produced alternative pronunciations outperform a single manually corrected one.

	Single	First	Second
top1	23.5	50.5	44.7

Table 3. WER of the three data sets, Single, First, Second. The top 1 baseform is generated by the L2S model and subsequently manually corrected.

4.2. Pronunciations generated with Subword Recognizer

We proceed, in this section, to report the results for the pronunciations generated with the subword recognizer described in Section 2.3. The pronunciations are generated from the first spoken utterances for all the words that were spoken twice. Hence, the words in the First set are presented to the subword recognizer and the generated top$_n \mid n = 1, 2..5$ baseforms *replace* the L2S seed baseforms of their corresponding words. The lexical baseforms of the Single set remain the L2S seed. The top 2 baseforms obtained with the subword recognizer are illustrated below:

```
yainnis : ( y uw n ax s | y uw n ax eh s td )
shawarma: ( sh w ao r m | sh w ao r m l ax s )
```

Since the baseforms are obtained from the First set, only the Single and Second sets are evaluated. As shown in Table 4, the top$_n$ WERs of the Second set are significantly improved as a consequence of these pronunciation-based baseforms. However, WERs for the Single set are consistently worse than before. One possible explanation for this degradation is the increased pronunciation confusion introduced by the subword recognizer.

	Single	Second
top1	27.8	45.9
top2	23.4	42.0
top3	20.1	39.8
top4	19.4	37.8
top5	19.1	37.3

Table 4. WER of the Single and Second sets as a function of the top$_n \mid n = 1, 2..5$ baseforms generated by the subword recognizer. The baseforms of Single remain the L2S seed.

Next, the spelling of each word in the First set is presented to the L2S model and a corresponding pronunciation lattice, K, is generated. K is used to constrain the search space of the subword recognizer. The resulting top$_n \mid n = 1, 2..5$ baseforms *replace* the L2S seed baseforms of their corresponding word. Similarly as before, the lexical baseforms of the Single

set remain the L2S seed. As illustrated below, the top 2 baseforms obtained with the constrained subword recognizer are clearly closer to the canonical pronunciations than the ones obtained with the unconstrained model.

```
yainnis : ( y ey n ax s | y ay n ax s )
shawarma: ( sh ax w ao r m ax | sh ao w ao r m ax )
```

Table 5 illustrates the WERs of the Single and Second sets as a function of the top$_n \mid n = 1, 2..5$ baseforms. Compared to the L2S seed baseforms, the top 1 WER for the Single set has an absolute deterioration of 0.8%, which is substantially better than the 2.1% deterioration obtained with the unconstrained subword baseforms. On the other hand, the absolute improvement for the Second set has dramatically increased from 1.9% to 12.2% absolute.

	Single	Second
top1	26.5	35.6
top2	22.1	34.0
top3	19.8	33.6
top4	19.3	32.4
top5	19.1	32.7

Table 5. WER of the Single and Second sets as a function of the top$_n \mid n = 1, 2..5$ baseforms generated by the spelling-constrained subword recognizer for words spoken twice.

4.3. Baseforms Combination

So far, we have replaced the L2S seed baseforms of the words in First with the ones acquired from the spoken utterances. We now proceed to combine the different acquired baseforms and report on WERs in Tables 6, 7, and 8. It is important to note, again, that whereas the Second lexicon has alternative pronunciations obtained from the spoken utterances, the Single lexicon does not. For example, if #total baseforms = 4 and # subword baseforms = 2, this implies that, for the Second lexicon, the last two L2S pronunciations (seed) are replaced with those obtained from the spoken utterances and the subword recognizer. On the other hand, for the Single lexicon, all 4 baseforms are from the L2S model.

Table 6 shows the WERs of the Single and Second sets as a function of both total number of baseforms as well as number of baseforms generated with the subword recognizer. The observed trend is for the WERs of Single and Second to decrease as the total number of baseforms is increased. However, for a fixed total number of baseforms, the performance of the Single set suffers while that of the Second set improves, as more baseforms are replaced with subword baseforms. This trend is consistent with the previously observed results in Tables 2 and 4 where the WER improves as the number of alternative pronunciations is initially increased. Furthermore, the increased pronunciation confusion introduced by the spoken utterances leads to performance deterioration for the Single set. We also note that the overall results are significantly better than those obtained with the L2S seed baseforms alone or the spoken utterances.

# total bf	# subword bf	Single	Second
2	1	22.2	33.8
3	1	20.1	32.2
3	2	20.8	32.8
4	1	19.7	29.9
4	2	20.1	31.2
4	3	20.1	31.2
5	1	18.8	31.6
5	2	19.3	30.6
5	3	19.7	29.9
5	4	19.9	29.9

Table 6. WER of the Single and Second sets as a function of combined baseforms. The first column is the total number of baseforms, and the second column is the number of subword baseforms for words spoken twice.

Table 7 exhibits similar behaviour as that in Table 6 except that the subword baseforms are generated with a spelling-constrained subword search space. We observe that combining the spelling-constrained baseforms to the seed baseforms does not result in as much gain as that reported in Table 6. One possible explanation is that the spelling-constrained baseforms are not very different from the seed baseforms, and hence do not introduce as much *new information* to the seed baseforms as the unconstrained subword baseforms.

# total bf	# constrained subword bf	Single	Second
2	1	20.8	34.4
3	1	18.1	34.0
3	2	19.3	33.3
4	1	17.9	35.3
4	2	18.9	34.0
4	3	19.0	32.2
5	1	17.9	35.2
5	2	18.5	35.1
5	3	19.2	33.6
5	4	19.1	32.2

Table 7. WER of the Single and Second sets as a function of combined baseforms. The first column is the total number of baseforms, and the second column is the number of spelling-constrained subword baseforms for words spoken twice.

Finally, Table 8 reports the best results obtained when the L2S baseforms are combined with both, the unconstrained and the spelling-constrained subword baseforms.

# total bf	# subword bf	# constrained subword bf	Single	Second
3	1	1	20.3	29.8
5	2	2	20.9	27.9

Table 8. WER of the Single and Second sets as a function of combined baseforms. The first column is the total number of baseforms, the second and third columns are the number of unconstrained and spelling-constrained subword baseforms for words spoken twice.

5. DISCUSSION AND SUMMARY

In this research, we have presented a new approach towards the automatic learning of lexical pronunciations. We have evaluated our approach on an isolated word recognition task for a 2k lexicon of restaurant and street names.

A linguistically-motivated CFG-based L2S model is used to learn the seed baseforms of the lexicon. To assess the performance of the L2S model, the top 1 L2S seed baseforms are manually corrected and evaluated. The modest improvement obtained with the manual modifications indicates the effectiveness of the L2S model. The lexical baseforms are then refined using spoken utterances of the lexicon, which are presented to a subword recognizer. Our best results are obtained when the L2S seed baseforms are combined with both spelling-constrained and unconstrained subword baseforms. To provide easy comparisons among the different experiments, we show in Table 9 the results for several experiments where the total number of baseforms is held constant at 3. For the Single set, the best result is for the L2S seed baseforms, and the least deterioration (0.2% absolute) is obtained when the seed baseforms are combined with the constrained subword baseforms. For the Second set, the constrained subword baseforms perform better than the unconstrained setup as well as the L2S seed baseforms. Furthermore, combining the three types of pronunciations provides the best results for the Second set and the best overall results.

Table	2	4	5	6	7	8
# L2S bf	3	0	0	2	2	1
# subword bf	0	3	0	1	0	1
# constrained bf	0	0	3	0	1	1
Single WER	17.9	20.1	19.8	20.1	18.1	20.3
Second WER	41.2	39.8	33.6	32.2	34.0	29.8

Table 9. Comparison of the WERs of the Single and Second sets as a function of baseforms. The first row refers to the Table number of the original experiment. The second, third, and fourth rows are the number of L2S, subword, and constrained subword baseforms respectively.

In this research, we have assumed perfect knowledge of the spelling of a word. In other scenarios, such as spoken dialogue systems, the user might provide a *spoken* rendering of the spelling of a word. We are currently investigating methods in which both the spoken word and spelling can be used to learn valid lexical baseforms.

6. REFERENCES

[1] "Web 1t 5-gram version 1," HTTP://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13.

[2] L. R. Bahl, S. Das, P. V. de Souza, M. Epstein, R. L. Mercer, B. Merialdo, D. Nahamo, M. A. Picheny, and J. Powell, "Automatic phonetic baseform determination," in *Proc. ICASSP '91*, Toronto, Canada, May 1991.

[3] B. Maison, "Automatic baseform generation from acoustic data," in *Proc. European Conf. on Speech Communication and Technology*, 2003.

[4] E. Fosler, M. Weintraub, S. Wegmann, Y. H. Kao, S. Khudanpur, C. Galles, and M. Saraclar, "Automatic learning of word pronunciation from data," in *Proc. Intl. Conf. on Spoken Language Processing*, Philadelphia, PA, Oct. 1996.

[5] T. Sloboda and A. Waibel, "Dictionary learning for spontaneous speech recognition," in *Proc. Intl. Conf. on Spoken Language Processing*, Philadelphia, PA, Oct. 1996, pp. 2328–2331.

[6] C. M. Westendorf and J. Jelitto, "Learning pronunciation dictionary from speech data," in *Proc. Intl. Conf. on Spoken Language Processing*, Philadelphia, PA, Oct. 1996.

[7] S. F. Chen, "Conditional and joint models for grapheme-to-phoneme conversion," in *Proc. European Conf. on Speech Communication and Technology*, Geneva, Switzerland, 2003.

[8] M. Bisani and H. Ney, "Open vocabulary speech recognition with flat hybrid models," in *Proc. Interspeech*, Lisbon, Portugal, 2005.

[9] L. Galescu and J. Allen, "Name pronunciation with a joint n-gram model for bi-directional grapheme-to-phoneme conversion," in *Proc. Intl. Conf. on Spoken Language Processing*, Denver, Colorado, 2002.

[10] B. Decadt, J. Duchateau, W. Daelemans, and P. Wambacq, "Transcription of out-of-vocabulary words in large vocabulary speech recognition based on phoneme-to-grapheme conversion," in *icassp02*, Orlando, Florida, May 2002.

[11] S. Seneff, "Reversible sound-to-letter/letter-to-sound modeling based on syllable structure," in *Proc. NAACL-HLT*, Rochester, NY, april 2007.

[12] G. Chung, C. Wang, S. Seneff, E. Filisko, and M. Tang, "Combining linguistic knowledge and acoustic information in automatic pronunciation lexicon generation," in *Proc. Interspeech*, Jeju, South Korea, Oct. 2004, pp. 328–332.

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

[14] L. Hetherington, "The mit finite-state transducer toolkit for speech and language processing," in *Proc. Intl. Conf. on Spoken Language Processing*, Jeju, South Korea, 2004.

[15] A. H. Gruenstein and S. Seneff, "Context-sensitive language modeling for large sets of proper nouns in multimodal dialogue systems," in *Proc. IEEE/ACL Workshop on Spoken Language Technology*, Palm Beach, Aruba, 2006.

[16] G. F. Choueiter, S. Seneff, and J. R. Glass, "New word acquisition using subword modeling," in *Proc. Interspeech*, Antwerp, Belgium, 2007.

[17] J. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, pp. 137–152, 2003.