Voice Building from Insufficient Data
classroom experiences with web-based language development tools

John Kominek, Tanja Schultz, Alan W Black

Language Technologies Institute
Carnegie Mellon University, Pittsburgh, USA
{jkominek,tanja,awb}@cs.cmu.edu

Abstract

To make the goal of building voices in new languages easier and more accessible to non-experts, the combined tasks of phoneme set definition, text selection, prompt recording, lexicon building, and voice creation in Festival are now integrated behind a web-based development environment. This environment has been exercised in a semester-long laboratory course taught at Carnegie Mellon University. Here we report on the students' efforts in building voices for the languages of Bulgarian, English, German, Hindi, Konkani, Mandarin, and Vietnamese. In some cases intelligible synthesizers were built from as little as ten minute of recorded speech.

1. Introduction

In the past decade, the performance and capability of automatic speech processing systems, including speech recognition and speech synthesis, has matured significantly. With the addition of machine translation linking input to output, the prospect of two people of different languages communicating together becomes a tantalizing possibility.

In light of the increasing trend towards Globalization, it has become important to support multiple input and output languages beyond the dominant Western languages (English, German, Spanish, etc.). Due to the high costs and long development times typical ASR, TTS, and MT, the need for new techniques to support the rapid adaptation of speech processing systems to previously uncovered languages becomes paramount [1].

The 3-year project and software toolkit known as SPICE[1] is an initiative intended to dramatically reduce the difficulty of building building and deploying speech technology systems. It is designed to support any pair of languages in the world for which a writing system exists, and for which sufficient text and speech resource can be made available. This is accomplished by integrating and presenting in a web-based development interface several core technologies that have been developed at Carnegie Mellon University. These include the Janus ASR trainer and decoder [2], GlobalPhone multilingual inventory and speech database [3], CMU/ Cambridge language modeling toolkit [4], Festival speech synthesis software [5] and FestVox voice building toolkit [6], Lexlearner pronunciation dictionary builder [7], Lemur information retrieval system [8], and CMU statistical machine translation system [9].

A new addition to this software suite is an embeddable Javascript applet that provides within-browser recording and playback facilities. By this means any two people in the word who would previously be separated by a language barrier can potentially speak with each other through our recognition/ translation/synthesis server (presuming access to a compliant Internet browser.) Also, our in-browser recorder provides a solution to an enduring problem of system development: namely, that of speech collection. It is no longer necessary that the system developer be able to locate native speakers of a particular language living nearby.

The SPICE software toolkit is in an early stage of development. To stress and evaluate the current state of the system, a hands-on laboratory course “Multilingual speech-to-speech translation” was offered for credit at Carnegie Mellon University. It ran for a single semester from January to May 2007, taught by three instructors: Tanja Schultz (ASR), Alan W Black (TTS), and Stephan Vogel (MT) [10]. All participants are graduate students studying language technologies. Students were paired into teams of two and asked to create working speech-to-speech systems, for a limited domain of their choosing, by the end of the course. The languages tackled were English, German, Bulgarian, Mandarin, Vietnamese, Hindi, and Konkani, a secondary language of India that does not have its own writing system but is transcribed through various competing foreign scripts. Interim results from this course are described in [11].

Here we report on the student's attempts in building synthetic voices in their language, including the role that pronunciation-lexicon creation played in their efforts. Creating a speech-to-speech translation system is a very ambitious task, meaning that only a portion of their time was allocated to TTS. Consequently, the students attempted to make good out of less material than is typical, i.e. much less than the one hour of speech of an Arctic-sized database [12]. Some hopeful students relied on less than 10 minutes of speech to build a voice from scratch – insufficient data, without doubt, hence the title of this paper.

2. TTS as a part of Speech-to-Speech

The speech synthesis system at the heart of SPICE is the CLUSTERGEN statistical parametric synthesizer [13], now released as part of the standard Festival distribution. We chose this technology because experience has shown that it (and the similar HTS [14]) degrades gracefully as the amount of training data decreases.

[1] Speech Processing – Interactive Creation and Evaluation Toolkit for new Languages.

Unlike the normal development procedure, however, in which the user executes a series of Unix scripts and hand-verifies the intermediate results, in SPICE all operations are orchestrated behind a web interface. The ASR and TTS components share resources. This includes text collection, prompt extraction, speech collection, and phoneme selection, and dictionary building. The interdependencies are depicted in below.

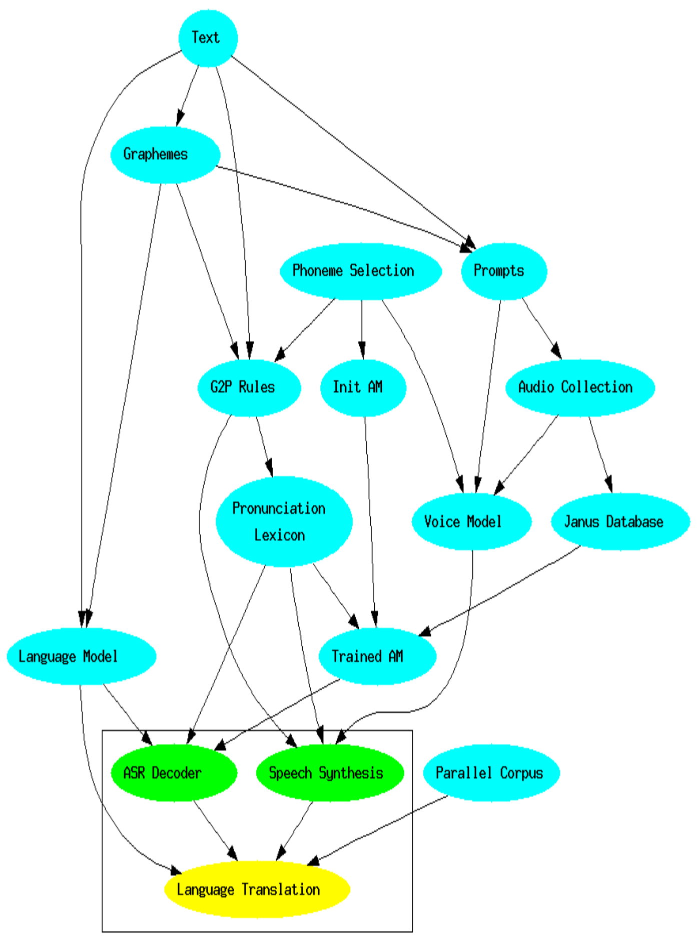
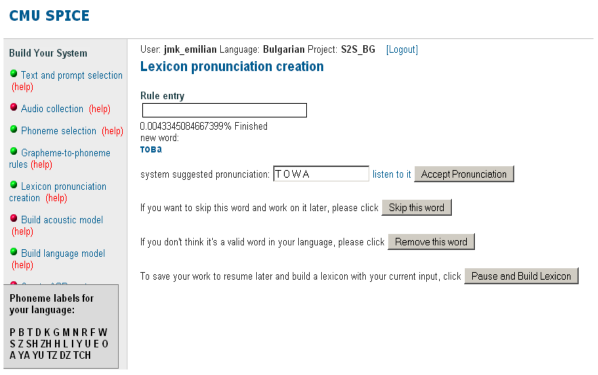


Figure 1. *High level component dependencies in the SPICE system.*

2.1. Development work flow – TTS

- **Text collection**. The development process begins with text collection. Text may be collected from the world-wide web by pointing the SPICE webcrawler at particular homepage, e.g. of a online news site. For additional control, the user can upload a prepared body of text. All of the students used this option, so that they could verify that the text is all within their chosen domain. Given the multitude of character encodings used worldwide, we imposed the constraint that the text had to be encoded in utf-8 (which includes ASCII, but not the vast population of 8-bit code pages.). Some began with large collections; others small, *cf.* section 3.
- **Prompt selection.** Typically one would convert the collected text to phonemes, then select sentence-length utterances that provide a balanced coverage of predicted acoustics, i.e. of diphones or triphones. At this stage though we have no means of predicting acoustics, and so prompts are selected on the basis of grapheme coverage.
- **Audio collection.** Depending on the source text, the prompt list is of varying length. We instructed the students to record at least 200 sentence-length prompts, adding material as needed if they had a shortfall.
- **Grapheme definition.** Once text is provided, the SPICE software culls all of the character and asks the user to define basic attributes of each. This includes class membership (letter, digit, punctuation, other), and casing.
- **Phoneme selection.** In this, perhaps the most critical stage, students select and name a phoneme set for their language. The interface assists this by providing a list of available phonemes laid out similar to the official IPA charts. An example wavefile is available for most phonemes so help in the selection. While this interface was intended to allow a phoneme set to be built up from scratch, not one student did that. Instead, they started from one or two reference lists and used the interface to make refinements.
- **G2P rules.** The development of grapheme-to-phoneme (or letter-to-sound) rules proceeds in a two-stage process. First, the user is asked to assign a default phoneme to each grapheme, including those that are unspoken (e.g. punctuation). This request required multiple explanations, as students tended initially to provide word sound-outs – declaring, for example, that 'w' is not associated with /W/ but is pronounced /D UH B AH L Y UW/ “double you”.
- **Pronunciation lexicon.** The second phase of G2P rule building goes on behind the scenes as the user accumulates their pronunciation lexicon. Words are selected from the supplied text in an order that favors the most frequent words first. Each word is presented with a suggested pronunciation, which the user may accept or manually correct. As an additional aid, each suggestion is accompanied by a wavefile synthesized using a universal discrete-phoneme synthesizer. Figure 2 shows a screen shot. After each word is supplied to the system, the G2P rules are rebuilt, thereby incrementally providing better predictions, similar to that of [15]. When the user is satisfied with the size of their lexicon the final G2P rules are compiled. These are then used to predict pronunciations for all the remaining lower frequency words.
- **Speech Synthesis.** With the necessary resources provided, the standard FestVox scripts have been modified to a) automatically import the phoneme set and look up IPA feature values, b) import the pronunciation dictionary, and c) compile the G2P rules into a transducer. The recorded prompts are labeled and utterance structures created. With this the language-specific F0, duration, and sub-phonetic spectral models are trained. The user can then test their synthesizer by entering text into a type-in box.


CMU SPICE

Build Your System

- Text and prompt selection (help)
- Audio collection (help)
- Phoneme selection (help)
- Grapheme-to-phoneme rules (help)
- Lexicon pronunciation creation (help)
- Build acoustic model (help)
- Build language model (help)

Phoneme labels for your language:

P B T D K G M N R F W
S Z SH ZH H L I Y U E O
A YA YU TZ DZ TCH

User: **jmk_emilian** Language: **Bulgarian** Project: **S2S_BG** [Logout]

Lexicon pronunciation creation

Rule entry

0.0043345084667399% Finished
new word:
това

system suggested pronunciation: T O W A listen to it [Accept Pronunciation]

If you want to skip this word and work on it later, please click [Skip this word]

If you don't think it's a valid word in your language, please click [Remove this word]

To save your work to resume later and build a lexicon with your current input, click [Pause and Build Lexicon]

Figure 2. *Web interface to lexicon builder.*

3. Descriptive Statistics

Prior to tacking their own language, students were encouraged to complete an English walk-though. The "English walk-through" is a prepared body of text and audio that is accompanied by step-by-step instructions. The walk-through material is based on the first 200 utterances of the rms Arctic database [12]. Having been designed for phonetic coverage and balance, the subset offered sufficient support for both ASR acoustic model adaptation and TTS construction; 200 utterances approximates the minimum required for success. In the tables below the figures for English are from this database.

3.1. Corpus Size

There are several measures when referring to the size of database using in voice construction. First is the original text corpus, from which the SPICE tools select a prompt list suitable for recording. Students are permitted to modify their prompt list, adding sentences they want covered in their domain and deleting those deemed inappropriate. During recording it is not unusual for additional prompts to be skipped (due to containing unpronounceable character sequences). As a rule of thumb, students were advised to record a minimum of 200 sentence-length prompts, even though the English voice was built from 96.

Language	***text corpus***			***selected prompts***		
		word counts			*word counts*	
	Utts	*types*	*tokens*	*Utts*	*types*	*tokens*
Bulgarian	23049	69607	508349	563	928	3517
English	200	798	1792	96	446	864
German	46328	49304	446765	435	1003	2913
Hindi	1543	558	12185	192	557	1524
Konkani	761	2008	3422	200	503	890
Mandarin	9925	22252	196120	199	1608	3669
Vietnamese	203	408	1520	203	400	1524

Table 1. *Size of language corpora in utterances and words (left), and of the selected prompt list (right).*

Language	***Prompts***	***Time***
Bulgarian	358	14:42
English	96	4:00
German	424	22:33
Hindi	191	9:51
Konkani	195	7:49
Mandarin	199	37:47
Vietnamese (rec)	203	10:41
Vietnamese (built)	77	3:38

Table 2. *Size of speech recordings (time in mm:ss). Whitespace was not trimmed from the prompts. Due to gaps in the lexicon, the Vietnamese voice was built from only a third of the available recordings.*

3.2. Word and character coverage

When building a voice for a new (i.e. previously uncovered) language, the development of a pronunciation dictionary is a major element of this task.. In the name of expediency, pronunciations for the 757 words needed for the English voice were extracted from CMU-DICT [16]. The students did not have his luxury and instead used the *lexlearner* component of SPICE to create a dictionary based on their supplied text. The one exception is Mandarin, for which the student uploaded a larger prepared dictionary.

Students were allowed to modify, supplant, and even replace the automatically selected prompts with their own list. This was true for Bulgarian, Hindi, and Vietnamese. Such allowances is a consequence of working in multi-purpose system: the language modeling component of ASR generally requires a large body of text, whereas TTS can often be improved if the text is targeted to the intended domain. The Hindi text for example was drawn from the Emille corpus [17], but the prompt list targeted the domain of cooking, restaurants and food recipes. In such cases there is a mismatch between the intended usage and assembled lexicon. Consequently the voice is forced to rely on grapheme-to-phoneme rules to "carry the day."

Language	***Dict***	***Text corpus***		***Selected prompts***	
	words	*types*	*tokens*	*types*	*tokens*
Bulgarian	396	0.57	49.36	0.0	0.0
English	757	95.55	99.77	100.0	100.0
German	1037	1.96	60.39	31.80	66.36
Hindi	356	64.03	86.87	0.0	0.0
Konkani	318	14.54	15.93	16.70	14.94
Vietnamese	288	70.34	59.54	1.25	0.46

Table 3. *Dictionary coverage of the original text corpus and selected prompts, in percent. For Bulgarian, Hindi,and Vietnamese the recorded prompt were not derived from the text.*

During construction of a pronunciation lexicon, it is the system that selects words and ask the user for the correct sequence of phonemes. Words are ordered from the most to least frequent. The benefit of this can be seen for Hindi, Bulgarian, and German. In German, the 1000 most frequent words is enough to cover 60% of the 450k text. However, coverage of the prompts is poor when the student opted not to go with the automatically selected list (breaking our original assumptions). Transcripts for the prompts then depend on the fidelity of grapheme-to-phoneme rules learned from a few hundred words. Since this is not optimal, a better alternative is to solicit coverage of the recorded prompts first, before proceeding onto the larger body of text if necessary.

Grapheme coverage may be even more important than word coverage, due to the fact that Festival will reject an entire utterance if it contains graphemes with an undefined pronunciation. This information is solicited during the development process (see section 2.1), but the system does not have checks in place to strictly enforce complete coverage. Oversights thus slip through, particularly when the student appends data to their text collection without revisiting the character definition protocol. The languages where this became problematic were Konkani (uppercase letters) and Vietnamese (various omissions).

Language	***Graphemes***	***Text corpus***	
	count	*types*	*tokens*
Bulgarian	74	85.14	99.81
English	51	100.0	100.0
German	53	100.0	100.0
Hindi	67	85.71	99.82
Konkani	52	57.69	93.14
Vietnamese	57	80.70	86.05

Table 4. *Grapheme coverage. Values are in percent.*

4. G2P Rule Learning

The complexity of the relation between graphemes and phonemes of a language of course varies dramatically from language to language. Of the languages described here, Bulgarian has the most straightforward, while English is highly irregular and Mandarin, being ideographic, exhibits no relation at all. Clearly, languages with a simple relation extrapolate more readily to unseen items, thus increasing the chances of a successful voice. And as previously pointed out, those projects with poor word coverage from the lexicon depend heavily on the G2P rules. From Table 3 these are Bulgarian, Hindi, Konkani, and Vietnamese.

The relative difficulty of languages (excepting Mandarin) can be seen by comparing the number of rules and rate of G2P rule growth with respect to the vocabulary size. These values are summarized in Table 5. Average letter perplexity – another indicative figure – is also included. As expected, English has the most complex G2P relationship. Bulgarian has a script that is nearly perfectly phonetic.

Language	***G2P Rules***			
	300 words	all words	*rules / letter*	*ave letter perplex.*
Bulgarian	51	54	1.019	1.002
English	360	727	25.07	3.350
German	236	523	4.023	1.932
Hindi	190	212	3.655	1.693
Konkani	205	223	7.964	2.356
Vietnamese	139	139	2.837	2.524

Table 5. *Comparison of G2P complexity. Note that Vietnamese is limited to 288 words.*

4.1. Case study: Hindi

To demonstrate the effort required to build a challenging lexicon, we report on the case of Hindi. Text was extracted from the Emille Lancaster Corpus [17], comprising 210 thousand words and 10.2 million tokens from the domain of current news. The Hindi speaker in the course used the SPICE toolkit to perform the following tasks: a) provide default letter-to-sound rules for each grapheme, b) provide pronunciations for the most frequent 200 words, c) correct automatically generated pronunciations for the next 200 words, and d) correct automatically generated pronunciations for the 200 words randomly selected from the remainder of the corpus. Error rates for these 200 words were then compared for three cases: a) G2P rules based solely on the default assignment, b) rules trained on the first 200 words, and c) words trained on the first 400 words. As summarized in Table 6, word accuracy increased from 23 to 41 to 51%. Additional detail is plotted in Figure 3.

G2P Rules		***Test Set***		
Training Size	Num Rules	1-200 words	201-400 words	400+ words
Default	49	52.7	32.3	22.6
LTS-200	127		51.0	40.9
LTS-400	216			50.5

Table 6*: Word accuracy on 187-word test set, for letter-to-sound rules based on 0, 200, and 400 training words (Emille corpus).*

The impact of these numbers can be seen in Figure 4, where projected token coverage of the Emille Hindi corpus is compare to the optimal coverage offered by a complete dictionary.

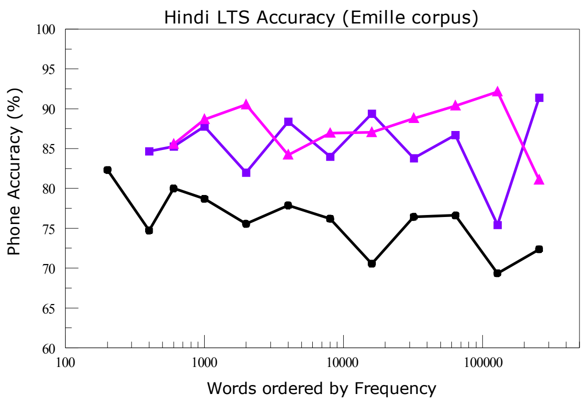


Figure 3*: Phone Error Rate of randomly sampled Hindi words taken from blocks on the log frequency scale. Each dot represents20 words.*

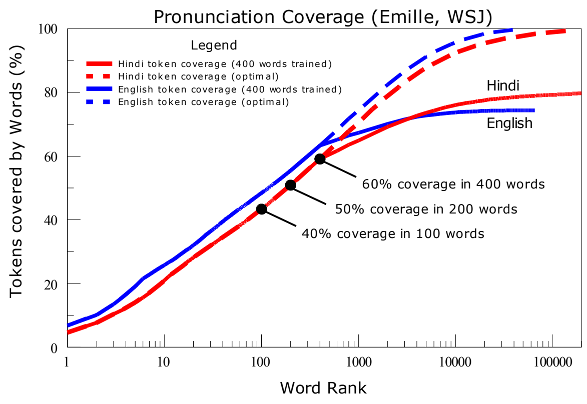


Figure 4: *Coverage of word tokens for Hindi when trained on the 400 most frequent words. A compatible curve for English is provided as a reference.*

5. Voice Quality Assessment

Each student was asked to provide an subjective impression of their voice, based on synthesis of in-domain prompts and of "random" things they think to type in. Three of the voices were a success, with the German voice receiving the most positive feedback – three people in the course speak German – though see section 5.1. The Hindi voices was also deemed good, though sentence-initial words tended to be confusing. The relative success for this pair can be attributed to reliable G2P rules, in order to go beyond the explicit lexicon (see Table 3). The English voice, built from just four minutes of speech was surprisingly understandable, though words outside the lexicon words were not uncommonly mispronounced.

The Vietnamese voice was poor – our two native speakers had trouble understanding what was said – though the tone contour was often correct. Since the Vietnamese voice was built from a mere three minutes of speech this result is understandable. Less understandable is the case of Mandarin, which for the amount data available *should* have been a good voice. We don't know yet whether this is attributable to errors in processing (i.e. bugs in the software), or some deep limitation confronted by tonal languages. The Konkani voice has been jokingly heralded as the best of its kind in the word (being the only one!) but is, the speaker admitted, incomprehensible. The details of why need to be determined. At this point we can safely conclude that 15% word coverage of the prompt list is insufficient for this language.

Language	***Time***	***Impression of Quality***
Bulgarian	14:42	(no feedback at time of writing)
English	4:00	understandable, mispronounces words
German	22:33	good, including prosody
Hindi	9:51	good, most words understood
Konkani	7:49	incomprehensible
Mandarin	37:47	fair
Vietnamese	3:38	poor

Table 7: *Impressionistic quality of voices as assessed by native speakers. For convenience the total length of each database is repeated from Table 2.*

5.1. Word comprehension

To establish a more quantitative assessment of intelligibility, we chose two of the better synthesizers for listening tests: German and Hindi. One of the German students provided transcriptions of the German voice. Twenty sentences were randomly selected from within the application domain and synthesized, with an additional four extracted from out of domain. The tester was allowed to listen to the synthesized sentences more than once, and to note which words became more clear after multiple listening. Transcripts were double-check for typographic errors. The Hindi listener was not a part of the class and thus was not familiar with the domain.
For the German in-domain sentences, 76% of the words were transcribed correctly, versus 55% for the out-of-domain. For Hindi the corresponding rates are very similar: 76% and 54%. Details are tabulated in Table 8.

Words correct (German)					
1-4	5-8	9-12	13-16	17-20	21-24
3/5	7/8	8/8	4/5	8/8	9/11
1/6	6/8	3/6	3/6	6/7	5/9
4/6	2/8	5/6	3/5	8/9	5/11
8/8	4/6	7/7	9/9	6/7	3/9
Words correct (Hindi)					
7/7	6/6	4/8	6/6	4/4	4/6
4/6	5/12	3/5	9/11	5/6	0/6
10/10	3/7	5/8	4/6	6/7	4/5
10/11	7/7	5/8	5/7	2/3	5/5

Table 8: *Words correct on randomly selected sentence for German (top) and Hindi (bottom). Sentences 1-20 are in-domain; 21-24 out of domain.*

6. Conclusions

By integrating ASR, TTS, and lexicon building into a single, simplified, web-based development framework, the aim of SPICE is to make speech technology available to developers that are not expert in language technology. Admittedly, the students participating in this lab do not fit the bill of *naive* users – our ultimate target audience. All are graduate students in the Languages Technology Institute and, due to the course credits on offer, were not just technically proficient but *motivated*. Their experience and observations has helped us identify deficiencies that need to be addressed before the software can reliably be employed by less sophisticated users – those that "just know their language."

For the task of voice building, more data-validity checks need to be incorporated. So that, for example, the user does not reach the end of a failed voice building attempt only to discover that the phoneme set, or character definition, or lexicon is in some ways deficient. In a similar vein: faced with the sizable task of providing pronunciations for thousands of words, our users have requested that they only be presented with the essential fraction, i.e. only words that the system is unsure about.

This raises a deep and challenging question: can the system be sufficiently "self-aware" that it knows when it needs more information, and when it can stop? At a practical level, we'd like the system to know when it has an amount of speech sufficient for building a good quality voice. Possibly we can resort to proxy measures of quality, such as mean cepstral distortion and average prediction error of prosody models. This may involve iterative cycles of feedback from the user to perform transcription tests and point out misspoken words. These remain open issues.

7. Acknowledgments

We thank Stephan Vogel for providing software and expertise in statistical machine translation. We also thank the following people for active participation: Nugyen Bach, Sameer Badaskar, Matthias Bracht, Joerg Brun, Roger Hsiao, Thuy Linh Nguyen, Kemal Oflazar, Sharath Rao, and Andreas Zollmann. Extra thanks go to Joerg Brun and Sameer Badaskar for transcribing test sentences. Sameer Badaskar also wrote the embedded recorder used for collecting speech. Matthew Hornyak kept all the various software in running order and deserves our undying gratitude.

This work is in part supported by the US National Science Foundation under grant number 0415021 "SPICE: Speech Processing Interactive Creation and Evaluation Toolkit for new Languages." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

8. References

[1] Schultz T. and Kirchhoff, K. (Eds.), *Multilingual Speech Processing,* Academic Press, 2006.

[2] Schultz T, Westphal M, Waibel A, *The Global Phone Project: Multilingual LVCSR with Janus-3*, 2nd SQEL Workshop, Plzen, Czech Republic, 1997.

[3] Schultz, T., *GlobalPhone: A Multilingual Speech and Text Database developed at Karlsruhe University.* ICSLP, Denver, CO, 2002.

[4] Clarkson P, and Rosenfeld R. *Statistical Language Modeling Using the CMU-Cambridge Toolkit,* Proceedings of ESCA, Eurospeech 1997.

[5] Paul Taylor, Alan Black, and Richard Caley, *The architecture of the Festival speech synthesis system,* Procedings of the Third ESCA Workshop in Speech Synthesis,1998, pp. 147–151.

[6] Black, A., and Lenzo, K., *The FestVox Project: Building Synthetic Voices*, http://festvox.org/bsv/ 2000.

[7] John Kominek, Alan W Black, *Learning Pronunciation Dictionaries: Language Complexity and Word Selection Strategies,* Proceedings of the Human Language Technology Conference of the NAACL, pp. 232-239, New York City, USA.

[8] The Lemur Toolkit. http://www.lemurproject.org.

[9] Stephan Vogel, Ying Zhang, Alician Tribble, Fei Huang, Ashish Venugopla, Bing Zhao, Alex Waibel, *The CMU Statistical Machine Translation System*, Proceedings of the MT Summit IX, New Orleans, USA, Sept. 2003.

[10] 11-733/735 Multilingual Speech-to-Speech Translation Seminar/Lab, http://penance.is.cs.cmu.edu/11-733.

[11] Tanja Schultz, Alan W Black, Sameer Badaskar, Matthew Harnyak, John Kominek, SPICE: Web-based Tools for Rapid Language Adaptation in Speech Processing Systesm, submitted to InterSpeech 2007.

[12] *John Kominek, Alan W Black CMU Arctic Speech Database, Speech Synthesis Workshop 5, Pittsburgh, USA, 2004.*

[13] Black, A., *CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling*, InterSpeech 2006, Pittsburgh, PA, September 2006.

[14] Heiga Zen, Keiichi Tokuda, Tadashi Kitamura, *An introduction of trajectory model into HMM-based speech synthesis*, Speech Synthesis Workshop 5, Pittsburgh, USA, 2004.

[15] Davel, M. and Barnard, E. *Efficient generation of pronunciation dictionaries: machine learning factors during bootstrapping*, ICSLP2004, Jeju, Korea.

[16] CMUDICT, http://www.speech.cs.cmu.edu/cgi-bin/ cmudict.

[17] Hindi EMILLE Lancaster Corpus, http://www.elda.org/ catalogue/en/text.