SPICE:
Web-based Tools for Rapid Language Adaptation in Speech Processing Systems

Tanja Schultz, Alan W Black, Sameer Badaskar, Matthew Hornyak, and John Kominek
InterACT, Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA
tanja@cs.cmu.edu

Abstract

In this paper we describe the design and implementation of a user interface for SPICE, a web-based toolkit for rapid prototyping of speech and language processing components. We report on the challenges and experiences gathered from testing these tools in an advanced graduate hands-on course, in which we created speech recognition, speech synthesis, and small-domain translation components for 10 different languages within only 6 weeks.

Keywords: Multilinguality, Language Adaptation, Multilingual Speech Recognition, Multilingual Speech Synthesis, Web Interfaces

1. Introduction

In the past decade, the performance of automatic speech processing systems, including speech recognition, text and speech translation, as well as speech synthesis, has improved dramatically. Propelled by the desire for ubiquitous information access, this has resulted in an increasingly widespread use of speech and language technologies in a wide variety of applications, such as voice-operated cell phones, car navigation systems, commercial information retrieval systems, and personal translation assistance. In light of the world's globalization, one of the most important trends in present-day speech technology is the need to support multiple input and output languages, especially when applications are intended for international markets and linguistically diverse user communities. As a result, new algorithms and strategies are required, which support a rapid adaptation of speech processing systems to new languages. Currently, the time and costs associated with this task is one of the major bottlenecks in the development of multilingual speech technology [1].

SPICE (**S**peech **P**rocessing - **I**nteractive **C**reation and **E**valuation Toolkit for new Languages), a three-year program sponsored by NSF, aims to significantly reduce the amount of time and effort involved in building speech processing systems for new languages. This was envisioned to be achieved by providing innovative methods and tools that enable users to develop speech processing models, collect appropriate speech and text data to build these models, as well as evaluate the results allowing for iterative improvements [2]. SPICE leverages the mature projects GlobalPhone [3] and FestVox [4], and implements bootstrapping techniques which are based on extensive knowledge and data sharing across languages, as well as sharing across system components. Examples for data sharing techniques are the training of multilingual acoustic models across languages based on the definition of global phone sets. Sharing across components happens on all levels between recognition and synthesis, including phone sets, pronunciation dictionaries, acoustic models, and text resources as displayed in Figure 1. Sharing with translation components will be implemented next.

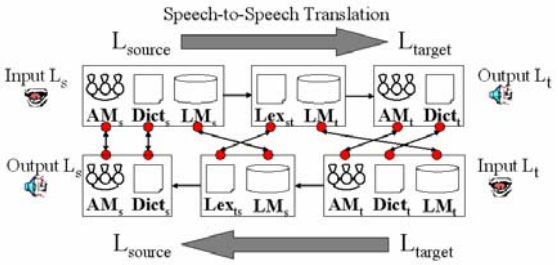

Figure 1*: Sharing across System Components*

In this paper we describe the design and implementation of these web-based SPICE tools, focusing on the user interface and lessons learned during a course taught at CMU with these tools (http://www.is.cs.cmu.edu/11-733).

2. Interface design and implementation

The SPICE interface has been designed to accommodate all potential users, ranging from novices to experts. Novice users are able to read easy-to-follow, step-by-step instructions as they build a language component. Expert users can skip past these instructions. In addition, file-uploading routines allow for feeding the bootstrapping algorithms with available data and thus shortcut the process. The result is that SPICE can collect information from the broadest array of people: a general audience of Internet users who may have little experience with speech tools, and a specific audience of speech and language experts, who can use data they already have.

2.1 Data harvesting and archiving

Our goal is to reduce the expense and expertise required to acquire and build systems from multilingual speech data. At present, there are only a few languages for which there are well-stocked repositories of text and speech data. Therefore, significant effort went into the design and implementation of web-based tools to perform automatic data harvesting. Users are also able to upload data they may have previously collected. If this is not available, the SPICE tools can be used to collect text from the web. To support audio collection the tools offer web-based recording facilities. As part of this, SPICE gathers information about the text and the speaker to properly annotate the data. Basic information about the language, such as its grapheme and phoneme set, is also solicited. This information is reused throughout the SPICE tools and attached to all other data

provided by the user. Information is preserved in standard formats. Text data is in UTF-8 and audio is single channel sampled at 16 kHz, 16 bit PCM, in Microsoft Riff format.

The collected data is archived such that it can be searched and retrieved by other users. Consequently, over time, SPICE users will create a repository of text and speech resources for many languages. Data in the repository will match the diversity of users, and also reflect the demand of each language. All data will be made fully accessible to users around the world.

2.2 Activity logging, profiles, projects

The SPICE tools are constructed around "projects" which contain all the information the user has provided about a given language. When users connect to the SPICE website, they enter their username, the language they are working on, and the name of the project. This allows users to pause and resume work on a project as they see fit. This "project" model also allows us to track usage of the SPICE tools on a per-user and per-language basis. We use this information to make improvements to the SPICE tools based on specific attributes of languages.

2.3 Web-based speech recording tool

Collection of good-quality speech data with transcripts is the first step in building or adapting acoustic models (AMs) for automatic speech recognition (ASR) and text-to-speech Synthesis (TTS). SPICE users are provided with a web-based facility to record speech data remotely over the Internet. It supports text data encoded in UTF-8 as this provides broadest coverage. The recorder is implemented in Java to enable it to work on a wide variety of operating platforms.

Once the user has provided a text corpus, a subset of prompts is selected automatically for recording. The selection procedure provides easy-to-read prompts that are phonetically balanced. Upon invocation, the web-recorder presents the selected prompts sequentially. Wavefiles are uploaded to the SPICE server once the user has verified the recordings are okay. Our web-based recorder presents multiple advantages: (1) rapid collection of large amounts of speech data by supporting collaborative work on common projects, (2) flexible collection by supporting multiple speakers and recording sessions, (3) integrity of the recorded speech data by requesting the user's consent and storing backup copies onto the user's system, and (4) uniform format of all recordings. In addition, the web-recorder reverts to offline mode when a network connection is lost. This facility makes data collection possible in areas of low or intermittent connectivity, enabling SPICE to reach geographically remote areas. Recording quality on a laptop can vary enormously. For ASR acoustic models, this recording environment may actually be similar to the end application environment thus such "noisy" data be useful, but for TTS we would like very clean recordings. At present we do not deal with these issues except in pointing them out to the users.

2.4 Component interfaces

The SPICE interface is organized around nine related tasks (soon to be ten with the addition of machine translation). As shown in Figure 2, these tasks are listed in a left hand column labeled "Build Your System". Users may move between tasks; SPICE has knowledge of component dependencies and allows a component to be active only if its prerequisites are satisfied. Online documentation is available for each component, as is a complete English walkthrough to explain the full process.

In the case of ASR acoustic model building, the process is organized as a sequence of six subtasks. Considering that this is a time intensive process, progress indicators are provided for all the intermediate training steps. Detailed log files from the intermediate steps may be examined at any time. Subtasks can be redone if necessary or, to shortcut the process, the user may also upload existing models. This flexibility is typical of how SPICE accommodates both novice and expert users.

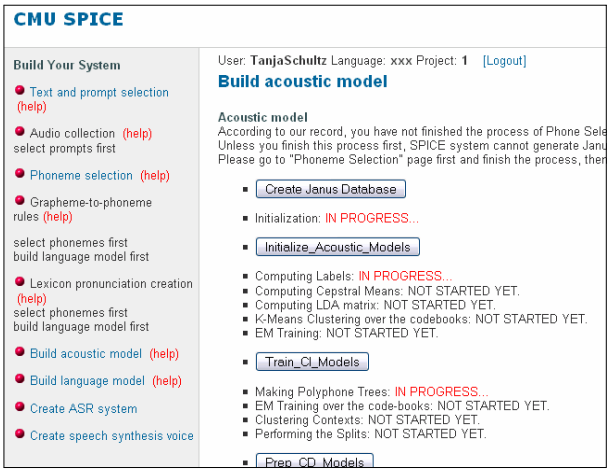

Figure 2*: Web-interface for ASR training*

The ASR development is built on the Janus Speech Recognition Toolkit (JRTk) using the IBIS decoder [5]. The training scripts are currently configured to train a tied 3-state HMM recognizer, where the number of triphone models and Gaussians per model is automatically adjusted according to the amount of training data. After bootstrapping AMs for the new language with the GlobalPhone MM7 multilingual acoustic model set [3], a context-independent and then a context-dependent system is created. These trained acoustic models are later used to generate forced-alignment data for building the TTS component.

2.5 Pronunciation dictionary construction

The pronunciation dictionary is essential to both ASR and TTS. In SPICE the Lexicon Learner component is responsible for eliciting pronunciations for words in the user's domain. It presents a sequence of words to the user, who provides the pronunciation as a sequence of phonemes in a phone set they define. The order of words selected from the supplied text is weighted according to token frequency. To reduce the difficulty of lexicon creation, each word is accompanied by a suggested pronunciation, along with a synthesized wavefile. The prediction is based on letter to sound rules that the system infers from the user's answers, which are updated after each additional word. The rules are seeded during an initialization stage in which SPICE asks the user for the phoneme most commonly associated with each letter. This is similar to the approach of [6]. Further details are described in [7].

2.6 Text-to-speech synthesis

The text-to-speech component offers a web interface to the underlying FestVox voice-building tools [4], and the CLUSTERGEN statistical parametric synthesizer [8]. These tools have been tuned to be effective for the relatively small amounts of speech data expected to be recorded. Following the approach of GlobalPhone we have investigated cross-language sharing of data to boost target language synthesizers [9].

3. Field experiences

While the web-based integration described here is new, the SPICE tools have been used previously to bootstrap speech recognition systems in Afrikaans [10], Bulgarian [11], and Vietnamese [12]. In [2] we described the development of a two-way speech translation system between English and Afrikaans within a 60-day timeframe. Currently, we are targeting the parallel development of speech processing components for a broader range of languages within a shorter timeframe. For this purpose we established a 6-weeks hands-on lab course at Carnegie Mellon University and simply adopted the native languages of all students who signed up for the course. This overcomes an important obstacle when developing speech processing systems, namely the lack of speakers and experts in the languages in question. Within the course the students were paired and asked to pick a limited domain topic and develop a simple speech-to-speech translation system. The goal for the student teams is to have them to talk to each other about their topic in their respective native languages. Students were required to rely solely on the SPICE tools and report back on problems and limitations of the SPICE system. These problems were then immediately tackled. This strategy allowed us to discover system and interface shortfalls, as well as detecting limitations resulting from lack of language support.

Many of the implemented features turned out to be quite helpful, sometimes in a surprising way. For example, one of the students is a native speaker of Konkani – the only one in Pittsburgh (population 2.3 million). However, as the web-recorder allows for collaboration, he conscripted friends and relatives back in his hometown in India to log-on to the SPICE interface and record native Konkani speech for his project.

3.1 Language peculiarities and challenges

During the lab course we have dealt with a considerable assortment of languages, simply by adopting the 10 native languages of our course students. These are Bulgarian, English, French, German, Hindi, Konkani, Mandarin, Telugu, Turkish, and Vietnamese. These exhibit a wide variety of features. For example, all writing system types occur in our selection: Mandarin uses the logographic Hanzi script; Bulgarian writes in Cyrillic; German, French and English use Roman; Telugu and Hindi use two different types of phonographic segmental scripts; Vietnamese is written in a phonographic featural script; and Konkani has no written form at all. Also, segmentation varies greatly between the languages: Chinese does not provide any segmentation, Vietnamese has segmentation but units surrounded by white spaces are not necessarily considered to be words. Morphology ranges from simple, low inflecting languages such as English to compounding languages, such as German, to highly agglutinating languages such as Turkish. Mandarin and Vietnamese are tonal languages. Bulgarian has stress that is unmarked in the orthography. Finally, the grapheme-to-phoneme relationship ranges from straightforward as in Turkish, to challenging as in Hindi (as the script does not reflect the order in which characters are pronounced), to difficult as in English. The Chinese script does not reveal any relationship, and Konkani lacks a script completely (thus we used a simple Roman script).

Meeting the needs of these languages required enhancements to the SPICE system. For example, to address the varying levels of segmentation, we modified the prompt selection system. To allow users to provide more detailed information about tonality and stress, we added additional input mechanisms to the phoneme selection system.

In summary, our current set of languages is varied in (1) the writing system used, (2) word segmentation, (3) morphology, (4) acoustic properties such as tonality and stress, (5) and letter-to-sound relationships. This variety has laid a good foundation for making the SPICE tools applicable to any human language.

3.2 Lexicon construction for Hindi

To demonstrate the effort required to build a challenging lexicon, we report on the case of Hindi, a language with which we do not have previous experience. Our text was extracted from the Emille Lancaster Corpus [13], comprising 210 thousand words and 10.2 million tokens from the domain of current news. A native of Bangalore and fluent speaker of Hindi 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. Each set of 200 words required between 35 and 40 minutes to complete. When measured on the held out selection of 200 test words (reduced to 187 after discarding as non-words), we found word accuracy more than doubled to over 50% with 400 training words.

Table 1*: Word accuracy on 187-word test set, for LTS rules based on 0, 200, and 400 training words (Emille corpus).*

	Rules	***1-200***	***201-400***	***401+***
Default	49	52.7	32.3	22.6
LTS-200	127		51.0	40.9
LTS-400	216			50.5

Based on the results of Table 1, we estimate that the LTS rules trained on 400 words will correctly cover 80% of the tokens (not word types) of the entire corpus. Reaching 95% token coverage is likely to require training on 2000 to 4000 words, i.e. about 12 hours of work, extrapolating from the hour and fifteen minutes spent on the first 400.

These estimates can be compared to English, for which CMUDICT is used as a reference [14]. To stay with news-oriented text we built LTS rules from the 400 most common words from the Wall Street Journal corpus of 1994-96 (66K in-vocabulary words, 43M tokens). As expected, the English LTS rules do not generalize as well as that of Hindi. However English makes greater use of high frequency words and so there is a

cross-over point at about 5000 words, with a final token coverage of 70%, while we estimate 80% coverage for Hindi.

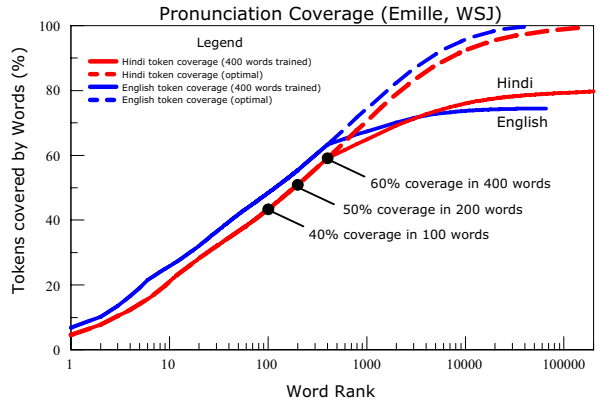

Figure 3*: Coverage of word tokens for English and Hindi when trained on the 400 most frequent words.*

3.3 Building end-to-end systems

Because ultimately we are interested in reducing the time required to create robust speech components, we are recording the amount of time spent on the various tasks. As of the time of writing, the lab course using the SPICE tools is half-finished. Partial results indicate that the bulk of up-front time is spent preparing the text and recording audio. Once the students have completed their projects, the distribution of time for all tasks can be analyzed. We are interested knowing the extent to which users go back and refine previous tasks rather than build an end-to-end system in a single straight-through shot.

Table 2*: Total time spent on each portion of the task*

Task	***Time Spent (hh:mm)***
Text collection	8:35
Audio collection	10:07
Phoneme selection	4:05
Language model building	1:25
Grapheme-to-phoneme specification	1:30

4. Conclusion

By gathering together and simplifying the interface, it is clear that the SPICE tools significantly reduce the time required to build usable speech recognition and speech synthesis models. With experience we are continuing to improve the components and their interfaces. Feedback from users and their experience will aid this process. Early indications with ten students in our current class are encouraging. Compared to previous student projects, we expect to see a substantial decrease in time required to completion.

Acknowledgements

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 NSF.

References

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

[2] Schultz, T. and Black, A., *Challenges with Rapid Adaptation of Speech Translation Systems to New Language Pairs*, ICASSP, Toulouse, France, 2006.

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

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

[5] Soltau, H., Metze, F., Fügen, C. and Waibel, A., *A one-pass decoder based on polymorphic linguistic context assignment*, IEEE ASRU Workshop, Madonna di Campiglio, Italy, 2001.

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

[7] Kominek, J. and Black, A,. *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] Black, A., *CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling*, INTERSPEECH-2006, Pittsburgh, PA, September 2006.

[9] Black, A and Schultz, T. *Speaker Clustering for Multilingual Synthesis*, MULTILING 2006, Stellenbosch, S. Africa.

[10] Engelbrecht, H. and Schultz, T., *Rapid Development of an Afrikaans-English Speech-to-Speech Translator,* Proceedings of International Workshop of Spoken Language Translation (IWSLT), Pittsburgh, PA, October 2005.

[11] Mircheva, Aneliya, Bulgarian *Speech Recognition and Multilingual Language Modeling*, Studienarbeit, Institut für Theoretische Informatik, Lehrstuhl Prof. Waibel, Universität Karlsruhe, March 2006.

[12] Viet-Bac, Le, Besacier, Laurent, and Schultz, Tanja. *Acoustic-Phonetic Unit Similarities for Context-Dependent Acoustic Model Portability*. Proceeding on Acoustics, Speech, and Signal Processing (ICASSP-2006), Toulouse, France, May 2006.

[13] EMILLE, http://www.elda.org/catalogue/en/text/W0038.html.

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

[15] Schultz, T., *Towards Rapid Language Portability of Speech Processing Systems.* Conference on Speech and Language Systems for Human Communication, Delhi, India, November 2004.