

# LANGUAGE-INDEPENDENT LEXICAL ACQUISITION

*Bertrand A. Damiba, Alexander I. Rudnicky*

School of Computer Science

Carnegie Mellon University

Pittsburgh, Pennsylvania 15213

## ABSTRACT

Lexicon construction is at the core of internationalizing speech systems, as it is the locus at which the correspondence between the written and spoken forms of a language is specified. For the most part, speech systems for a given language benefit from the attention of native speakers and the opportunity to tune performance over time, allowing the cost of lexicon development to be amortized over time. On the other hand rapid deployment of recognition capability for new languages stresses the need for rapid availability of a usable lexicon.

We propose a decomposition of the lexicon building process, into four discrete and sequential steps that simplify and speed up the creation of language knowledge bases for recognition and synthesis. Results from four languages are discussed.

## 1. INTRODUCTION

Many interesting questions arise when adapting existing speech systems to languages other than the original target language [12]. Most of the assumptions that have found their way into the core of single language designs do not necessarily hold when applied to other languages. For they are often expressed with different character sets, have different phoneme sets and pronunciation rules along with other specificities. Moreover, native speakers of the language would have tuned performance over time.

A number of approaches have been proposed. Some advocate completely rebuilding systems for a new target language [9][8]. Others prefer rebuilding new statistically based systems aimed at cross language portability [5]. Although valuable, these approaches can be very costly and result in redundant work. When dealing with rapid deployment systems, an approach that would make the most use of existing systems and require smaller scale commitment would be, perhaps, better suited.

We have been exploring problems of automatic speech recognition and text-to-speech synthesis portability in the context of the DIPLOMAT speech-to-speech translation project [6], successfully dealing with Serbo-Croatian, Haitian Creole and Korean languages.

In speech technology terms, a language mostly finds its uniqueness in the way it sounds and in its script, both of which are specified in its lexicon. The lexicon is the most localized part of any speech system, since, once the character set and pronunciation issues are solved many of the other components of a system need no further internationalization.

Some other approaches strongly rely on machine learning [11], and are therefore dependent on the amount and quality of ex-

isting data, an assumption that doesn't hold for many languages. Of the existing approaches to language-independent phoneticizing grammars [7][15], most do not consistently address the character set issue, neither do they offer grammars that are legible by non-linguistically trained experts. Our approach relies on the availability (or tele-availability) of a native informant and their knowledge of the language. We do not however assume that this needs to be someone with formal training in linguistics or speech recognition, only that they possess a basic familiarity with computers.

We describe a language-independent phoneticization process that consists of four successive transformations, described in the next section. This process transforms a Unicode string into its corresponding phonetic string, solving the character set issues along the way.

## 2. PHONETICIZATION IN FOUR STEPS

The four steps consist of a Unicode transliteration followed by normalization of the orthography, a phoneticization of the normalized string and a final phonological pass. Each discrete step has a well-defined goal, which is simple enough to potentially open up the process to non-linguistically trained users. Unlike machine learning-based approaches [4], our ultimate aim is not to completely automate the lexical acquisition process, but rather to structure it in a way that will allow native speaker (not necessarily a linguist, though computer literate) to create a knowledge base for a new language.

Rule collisions are often the single obstacle in the successful rule-based phoneticizing of a language [15]. By partitioning the rule space into three discrete sections (PLI sections 2-4), the user only needs to ensure that the rules created are consistent within the space they address. We believe that the resulting reduction in ruleset complexity simplifies rule management issues and speeds up the prototyping process.

**PLI#1: Transliterating from Unicode to ASCII.** As Unicode has become the commonly accepted standard universal character set, it opens our process to most known languages. Unicode is a fixed size character set, where each text element is encoded in 16 bits (UCS-2); this allows uniformity across languages. More importantly, the Unicode consortium [14] has set standards for the processing of many scripts that defy the assumptions made by ASCII (i.e. bi-directional algorithms, Hangul syllable decomposition/ composition algorithm etc.) and can be helpful to speech technology. For these reasons, Unicode must be included in any attempt at language independent lexical acquisition.

The process of transliterating from Unicode maps each relevant Unicode code point used in the target language to an ASCII

string to be used in the later steps. In some language one text element encodes more than one linguistic phenomenon. For example, in French vowels often carry diacritical marks, in Hangul where each text element is a syllable, a text element may carry up to four jamos (that is, single letter of the Korean script). Transliteration allows us to create our own, string-based internal character, well suited for phonetic processing which has the virtue of fitting with existing ASCII-based ASR systems.

Speech technologies often exploit the relationship between the spoken word and its written representation, yet not all languages have a phonetic script (Mandarin, Cantonese). Transliterating allows us to recreate the script at the text element level, and recreate, for the purposes of the task at hand, that crucial relationship between the written word and the spoken word.

**PLI#2: Standardizing the orthography.** Languages carry in their orthography a certain complexity as a result of their history. Often the orthography to sound relationship is not quite intuitive (i.e. English: "knight" sounds more like "nite", in French "paon" sounds more like "pan", in Korean: f [ㅈ]ㅇ sounds more like 가 [ㄱ]g). Other languages are quite flexible in the way they are written, allowing several orthographies for the same word (in Haitian Creole "pwezidan" and "presidan"). In the case of homophones, the orthography marks a semantic difference (i.e. English "know", "no"). This creates the need for a phonetic standardization of sorts for the purposes of speech technology, where often these artifacts are obstacles to that important script to sound relationship.

This step also gets us closer to a context independent pronunciation of subword units, alleviating the load on the subsequent phoneticization steps [7].

**PLI#3: From graphemes to phonemes.** Given a standardized orthography, this step implements basic grapheme to phoneme mapping. All the remaining context dependent-pronunciation combinations are addressed during this step. Phoneme interaction such as nasalization need not be created here in order to reduce collisions.

**PLI#4: Phonological processes.** In any given language, regardless of the orthography some pronunciation rules are solely based on sound. This section is also meant for differentiating between allophones, depending on their phonetic context.

The system is currently implemented in several components: a simple rule grammar that specifies transformations, an International Phoneticizing Engine (IPE) that applies these rules to a lexicon, and under development, a GUI-based development environment that allows designers to create and manage PLI rulesets. A more detailed description is provided in [3].

## 3. EVALUATION

### 3.1 Creating PLIs for Korean, Haitian Creole, English and Serbo-Croatian

To understand the above process, we created PLI rule sets for 4 different languages. The most frequent words of our text corpora were chosen to create PLIs. The assumptions were that,

most pronunciation phenomena would be encountered in that subset and secondly, the coverage that these high frequency words have on the language would result in a better performing PLI when applied to natural text.

**Haitian Creole:** Haitian Creole is a language for which standardized orthographic representations have only recently been proposed. While the relationship between the sound of the language and the script is quite simple, one can find many versions of the same word. For our experiment, 600 words were used to build the PLI. Created by one native French speaker with the collaboration of a fluent Creole speaker.

PLI#2 rules tend to standardize the Haitian Creole orthography. Since most words are written much the way they sound, little else is needed. It might be noted that Haitian Creole has many loan words in its lexicon that defy its own pronunciation rules. These are entered as exceptions in this section. Our text corpus, obtained through the translation by 3 native speakers, didn't offer the variety of orthographies that can be found in the language at large; for which more standardization rule would probably be required.

**Korean (Hangul):** Korean presents an interesting challenge because it has a very large and very different character set from English, yet is a phonetic script requiring further rules. A total of 900 words were used in building the PLI. Two native speakers of Hangul did the work.

In the PLI#1, Unicode Hangul phonetic syllable was mapped by rule to a corresponding string of three jamos. The 11,179 code points were then mapped automatically.

Hangul is a regular language, where each jamo has a pronunciation depending on its position in the 3-jamo wide syllable. There are some inter-syllable interactions that affect the pronunciation; these rules were expressed in PLI#2. In particular, some composed jamo (where one jamo represents the ligature of two other) have pronunciations entirely dependent on their context. PLI#4 implements some nasalization and deletion rules

**English:** A native speaker and a fluent non-native speaker together created the PLI; 1030 words were used.

PLI#2 contains 220 word exceptions and 286 standardizing rules: 506 total rules. It took 25 man-hours to create. Although PLI#2 is advised to be used for rewriting a word in an orthography that is more phonetically correct, while developing a PLI for English, we used phonemes in that same section as a simple mechanism for blocking further transformations. Of the 286 standardizing rules, 47.5% contain phonemes. An alternative would have been to add the affected rules as exceptions; however this would defeat the goal of creating a general ruleset capable of correctly transforming unseen words. Another approach uses a prior syllabification step to alleviate this very problem, thus limiting undesirable application of rules at these boundaries [6]. We believe that because the consistent spotting of syllables can be illusive to non-linguistically trained users, this step didn't fit in a system ultimately meant to open the phoneticization process to non-linguists.

**Serbo-Croatian:** Created by a linguist (non-speaker of Serbo-Croat) with the help of linguistic reference manual. Like Haitian Creole, Serbo-Croatian is written very much like the way it

sounds, allowing most of the transformations to be handled in PLI#1 and PLI#3.

Language	PLI#1	PLI#2	PLI#3	PLI#4
English	54	506	47	63
Korean	11,769	240	1	10
Haitian Creole	100	7	45	5
Serbo-Croatian	51	1	30	0

Table 1. Rules needed for each PLI section across languages

**Discussion:** The experience of building PLI files for these different languages led to some insights into the process. For example, PLI#1 should be by nature expansive, whether in importing archaic syllables for Hangul, or dealing with ligatures in Serbo-Croatian. Because PLI#1 defines the code space of the Unicode text corpus, flexibility is key. It is probably the most important section for scripts that defy many of the assumptions made by ASCII (e.g. Hangul), because the way the code points are imported will affect the complexity of the following sections.

Lexicon	% WER
CMU Dictionary, version 0.5b	12.1
IPE English	30.8
IPE English, modified (*)	20.1

Table 2. Error using English PLI vs. CMU dictionary 0.5b. (\*)10 word entries were changed; three exceptions {Kansas, saint, arriving} were added and 7 high-frequency words {the, to, what, a, for, from, are} were given alternate pronunciations.)

PLI#2 serves many purposes, most importantly in removing ambiguities in the sound to script relationship. It presented itself as a real challenge for English, a language rich in exceptions with a weak script to sound relationship. Standardizing the orthography is the most laborious task of extracting phonemes from words. Because of the large number of rules in this section, some rule collision occurred despite the division of the rule space in three distinct subspaces. In languages with a good script to sound relationship (Haitian Creole, Serbo-Croatian), the PLI#2 was used to respell foreign words. In the case of Haitian Creole to standardize a language that allows multiple orthographies for the same word in order to deal with the same flexibility the language allows.

In PLI#3 for all languages mapped graphemes or sequences of graphemes to a phoneme. Across languages it was the simplest section to create for it is based on the assumption that, after PLI#2, the language's script is standardized.

PLI#4 across languages dealt with phonological phenomena (stress, nasalization etc.), but also served as a "clean-up" section where improbable phoneme sequences were reassessed.

More generally, the PLI grammar should ideally allow multiple pronunciations, as they are necessary in practice (see below), while not compromising simplicity and legibility.

In addition many languages have their own morphological identities, in which the assumption that a sentence is a sequence of discrete words separated by spaces doesn't hold (Thai, Farci). This illustrates the need for a Unicode-based morphological tagger that would extract morphemes based on a base dictionary with features.

### 3.2 Evaluating English and Korean PLIs

It is possible to assess the accuracy of a PLI-based lexicon by comparing it to a handcrafted version of the same lexicon. We did this for the English PLI by comparing it with a well-established pronunciation dictionary of American English developed at Carnegie Mellon [13].

We used the 1993 ATIS evaluation test set, with a 2997 word dictionary. The test set includes 68 speakers and 136 utterances and has a perplexity of 52.8. The comparison was performed using context-independent phonetic models (see Table 2). We believe the PLI-based approach produces reasonable performance for a rapid-deployment system. Minimal tuning significantly reduces error.

Using existing tools [2][10] we also evaluated recognition performance for a Korean PLI. The text corpus used for both the language model and the speech data collection corpus was obtained from publicly accessible web sites. The text (originally in KSC Wansung encoding) was converted to Unicode (UCS-2) and segmented into sentential utterances. The utterances were phoneticized. We then used a minimum preserving scheme to extract a diphonic rich subset of utterances to populate the recording script.

The training data consisted of 21 hours of speech read by 162 (70 female and 92 male) native Korean speakers. The pronunciation dictionary was generated with the aforementioned Korean PLI and was used for both the training of the acoustic models and the recognition tests.

	Trigram	Bigram	Unigram
Word Error (%)	8.45	15.67	25.25
Syllable Error (%) (*)	5.54	9.70	16.61

Table 3. Error Rates using Korean PLI. LM text corpus size: 14358, Unique words in transcript: 310, Dictionary size: 8550 words. (\*) In written Korean, morpheme boundaries are looser than in English (i.e the word "스핀아웃현장" can also be correctly written as the sequence of the words: "스핀", "아웃" and "현장"), therefore word error rates for Korean systems can be pessimistic. As an informative counter-balance, syllable error rates are given here as an optimistic assessment of the recognition results.

The speakers used in the recognition run (1 female and 1 male) were not included in the training. The test corpus contains 100

utterances (13.1 minutes of speech) uttered by speakers not used in the acoustic training. Results are in Table 3.

### 3.3 An English vs. Korean comparison

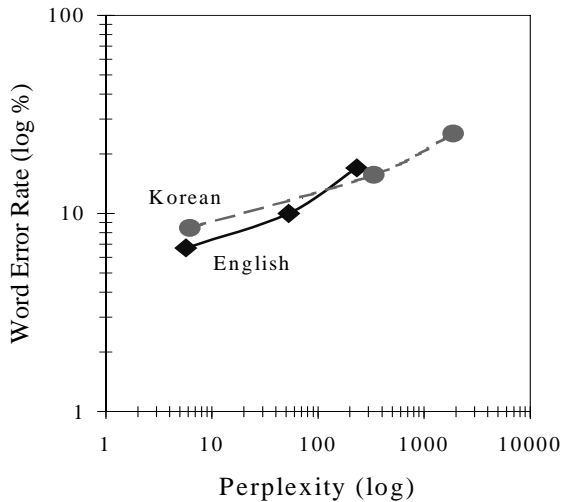


Figure 1. Effect of language model perplexity on the Word Error rates for Korean and English recognizers.

We compared Korean context-dependent acoustic models, with established English context-dependent acoustic models (from the ATIS domain), using language model perplexity as the decaying factor. As we can see in Figure 1, the overall error patterns for English and Korean are similar. We can however note that for perplexities below 100 the English acoustic models fare better. But beyond perplexities of 100, the Korean acoustic models seem to do even better than the English models. In our pronunciation dictionaries, Korean words tend to be phonetically longer than English words (8.31 phones/word for Korean and only 6.24 phones/word for English). The reduced phonetic confusability of the Korean lexicon may account for the error trend.

Most importantly these results show that the Korean system behaves similarly to a comparable English system. Since the Korean acoustic models were built using exclusively a pronunciation dictionary automatically created by the Korean PLI and a conventionally developed English system, the proposed phoneticization process appears to be a viable prototyping technique.

## 4. CONCLUSION

The PLI grammar and its interpreter enables rapid internationalization of speech systems. Simplicity and legibility were the guiding principles of their design since the ultimate goal is to also allow non-linguistically trained subjects to create PLIs. We found that this approach was suitable for several different languages and that it resulted in usable recognition performance. While this process allows us to successfully prototype a lexical knowledge base for a new language, it may need to be augmented to support a refinement process (or simply reserved for rapid prototyping and text-to-phone generation).

## 5. ACKNOWLEDGEMENTS

We would like to acknowledge the valuable advice and contributions from Joseph Kenney, Dr. Suk-Dong Kim, Scott Hansma, Eric H. Thayer, Dr. Mosur Ravishankar, Dr. Roni Rosenfeld and Photina Jang.

## 6. REFERENCES

- [1] Carlson Ron, Bjorn Granstorm (1976), "Text-to-speech Based Entirely on Rules", ICASSP '76, 686-688.
- [2] Clarkson, P., Rosenfeld, R. (1997), "Statistical Language Modeling using the CMU-Cambridge Toolkit", EUROSPEECH '97, 2707-2710.
- [3] Damiba, B.A. and Rudnicky, A.I. (1997) Internationalizing speech systems through language independent lexical acquisition. Carnegie Mellon University School of Computer Science Technical Report 97-186.
- [4] Damper R.I. (1995), "Self-learning and connectionist approaches to text-to-phoneme conversion", in Connectionist Models of Memory and Language, Levy J., Bairaktaris J., Bullinaria J., and Cairns P. (eds.), UCL Press, London, 117-144.
- [5] Deng, L. (1997), "Integrated-multilingual Speech Recognition using Universal Features in a functional Speech Production Model", ICASSP '97, 1007-1010.
- [6] Frederking, R., Rudnicky, A., Hogan, C., (1997) "Interactive Speech Translation in the DIPLOMAT Project", Spoken Language Translation Workshop ACL- '97.
- [7] Hertz, Susan (1982), "From text-to-speech with SRS", Journal of the Acoustical Society of America, 1155-1171.
- [8] Lee Lin-Shan, Chiu-Yu Tseng, Hung-Yan Gu, F.H. Liu, C. H. Chang, S. H. Hsieh and C. H. Chen (1990), "A Real-time Mandarin Dictation Machine for Chinese Language with Unlimited texts and very large Vocabulary", ICASSP '90, 65-68.
- [9] Matsuoka Tatsuo, Katsutoshi Ohtsuki, Takeshi Mori, Sadaoki Furui and Katsuhiko Shirai (1996), "Large-Vocabulary Continuous-Speech Recognition Using a Japanese Business Newspaper (NIKKEI)", Proc. Of the ARPA Workshop on Spoken Language Technology, Austin TX, Morgan Kaufmann, Cohen, Ed.
- [10] Placeway, P. et al (1997), "The 1996 Hub-4 Sphinx-3 System", Proc. DARPA Speech Recognition Workshop.
- [11] Sejnowski, T.J. and Rosenberg, C.R. (1986), "Nettalk: a parallel network that learns to read aloud" The Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01.
- [12] Steeneken, H.J.M. and Lamel, L.F. (1994) "SQUALE: Speech Recognizer Quality Assessment for Linguistic Engineering", Proceedings ARPA Workshop on Spoken Language Technology, Plainsboro, New Jersey.
- [13] The CMU Dictionary: <http://www.speech.cs.cmu.edu/comp.speech/Section1/Lexical/>
- [14] The Unicode Consortium (1996), "The Unicode Standard, version 2.0", Addison-Wesley Publishing Company.
- [15] Van Coile, Bert M.J. (1989), "The DEPES Development System for Text-to-Speech Synthesis", ICASSP '89, 250-253.