Exploring minimal pronunciation modeling for low resource languages

Marelie Davel[1]*, Etienne Barnard*[1]*, Charl van Heerden*[1]*, William Hartmann*[2]*,*
Damianos Karakos[2]*, Richard Schwartz*[2] *and Stavros Tsakalidis*[2]

[1]Multilingual Speech Technologies, North-West University, Vanderbijlpark, South Africa.
[2]Raytheon BBN Technologies, Cambridge, MA 02138, USA.

marelie.davel@nwu.ac.za, {whartman, dkarakos, schwartz, stavros}@bbn.com

Abstract

Pronunciation lexicons can range from fully graphemic (modeling each word using the orthography directly) to fully phonemic (first mapping each word to a phoneme string). Between these two options lies a continuum of modeling options. We analyze techniques that can improve the accuracy of a graphemic system without requiring significant effort to design or implement. The analysis is performed in the context of the IARPA Babel project, which aims to develop spoken term detection systems for previously unseen languages rapidly, and with minimal human effort. We consider techniques related to letter-to-sound mapping and language-independent syllabification of primarily graphemic systems, and discuss results obtained for six languages: Cebuano, Kazakh, Kurmanji Kurdish, Lithuanian, Telugu and Tok Pisin.

Index Terms: spoken term detection, graphemic systems, pronunciation lexicons

1. Introduction

As the availability of speech training data has increased, so has the popularity of graphemic lexicons. Without explicit grapheme-to-phoneme (G2P) conversion, speech recognition systems developed using graphemic lexicons link the acoustic realization of a word directly to its orthography. These systems are of particular use when developing speech processing systems for the less-resourced languages of the world, for which extensive pronunciation lexicons are typically not available. Graphemic systems are therefore used in two very different scenarios: one which is well-resourced to the extent that sufficient samples of most words (or sub-words) are directly observed during training, and another where the language is under-resourced: pronunciation lexicons are not available and training data is limited. We analyze the latter case.

Between fully graphemic and fully phonemic systems, there exists a continuum of options. Here we investigate minor modifications to a graphemic system that improve accuracy without requiring significant effort to design or implement. We investigate four aspects: (1) dealing with known language-specific spelling idiosyncrasies; (2) harmonizing unit definitions across languages in order to support cross-lingual data sharing; (3) reducing the prevalence of rare units, which are otherwise poorly modeled; and (4) creating syllable-like units for sub-word modeling purposes. Analysis is performed in the context of the IARPA Babel program.

2. Background

The aim of the Babel program, sponsored by the US Intelligence Advanced Research Projects Activity (IARPA), is to support the rapid development of automatic speech recognition (ASR) and spoken term detection (STD) capabilities in new languages. To this end, participants are provided with limited amounts of orthographically transcribed speech, along with additional untranscribed speech and additional linguistic information such as pronunciation lexicons and basic linguistic facts (dialect distributions, phoneme and grapheme sets, etc.) These facts are provided in *Language-Specific Peculiarities* (LSP) documents, provided for all Babel languages.

In total, the Babel project will investigate 26 different languages over a period of four years; in the current paper we consider the six languages released at the beginning of year three, namely Cebuano (ceb), Kazakh (kaz), Kurmanji Kurdish (kur), Lithuanian (lit), Telugu (tel) and Tok Pisin (tpi). On an annual basis, participants use the transcribed training and development data provided, as well as a set of development keywords, to produce STD results. Results are verified during the evaluation period, using a new set of evaluation keywords and untranscribed evaluation audio made available to all participants. Since 2015, the manually produced pronunciation lexicon included in the provided resources may no longer be used for official submissions to the Babel challenge.

Babel allows for various training and testing conditions; in order to understand the issues relevant to a severely under-resourced language, we limit our attention to the so-called *Very Limited Language Pack* (VLLP) condition, in which only three hours of transcribed speech data is available per language. We use the official training and development sets provided by NIST.

For ASR and STD, a state-of-the-art system, as described in [1], is employed. It uses stacked bottleneck features, discriminatively trained multilayer perceptron classifiers and innovative keyword-spotting techniques to obtain consistently competitive results in evaluations to date. When combining whole word systems with sub-word systems, we use the techniques described in [2].

The idea of using graphemes for ASR was proposed in [3], and shown to be competitive on several different tasks in subsequent research [4, 5]. The growing interest in under-resourced languages during recent years [6] has lead to a renewed interest in grapheme-based ASR, and below we investigate how such systems can be improved with different amounts of language-specific information.

3. Approach

Our aim is to develop a pronunciation lexicon given only a training vocabulary and minimal language-specific information, as captured in the Babel LSP documents. We aim to explore only those techniques that can be applied quickly and with limited human effort, and focus on four aspects: (1) harmonizing unit definitions across languages; (2) modeling known language idiosyncrasies through rewrite rules; (3) reducing the prevalence of rare units; and (4) language-independent syllabification. Together, these provide a simple but fairly effective modeling strategy when developing an ASR or STD system with limited resources.

3.1. Harmonizing grapheme definitions

At least a minimal degree of cross-lingual harmonization of acoustic units is required in order to support cross-lingual data sharing. For the VLLP condition, multilingual features are typically extracted using data from different languages, and these features then tuned to the target language. As a similar phoneme can be represented by quite different grapheme combinations per language, consistency across languages is achieved by using grapheme-to-phoneme mappings, rather than grapheme-to-grapheme mappings. For example, in Table 1 the most typical realizations of the letter 'o' are shown, as well as some of the most typical source letters for the phonemes /o/ and /o:/, using SAMPA notation. When data from different languages are combined, the phonemes are matched, rather than the (potentially acoustically dissimilar) graphemes. Note that this change in representation does not affect mono-lingual systems: when considered individually, these remain close to graphemic.

Table 1: *Examples of one-to-one G2P maps.*

lang	*letter*	*phone*	*lang*	*letter*	*phone*
ceb	o	o	tel	ఒ	o
kaz	o	uU	tel	ఓ	o:
kur	o	o	lit	o	o:

3.2. Modeling language idiosyncrasies

When mapping each letter to its default phoneme, it is clear that many languages contain minor idiosyncrasies that can be modeled using straightforward G2P rewrite rules. To illustrate, we list examples of many-to-one, one-to-many and many-to-many G2P rules for Cebuano in Table 2. For the current analysis, these rules were obtained exclusively from the LSPs, resulting in an extended set of G2P rewrite rules per language, described in more detail in Section 4.1.

Rewrite rules in the form of direct maps are not sufficient for all languages. For example, Telugu is a syllabic language where the orthography does not map to acoustic units in a natural way. Specifically, consonants are associated with an inherent vowel, and how the vowel is realized can be changed by diacritics occurring close to the consonant. Modeling this phenomenon explicitly leads to better performance, as discussed in Section 4.4.

Table 2: *Cebuano examples of multiple grapheme to multiple phoneme mappings.*

many-to-one			*many-to-many*			*one-to-many*		
t h	→	t	- a	→	? a	X	→	k s
n g	→	N	- I	→	? I			
t i y	→	tS	- u	→	? u			

In most languages, a special category of problematic words are letter-based abbreviations ('spelled words') such as 'SMS' or 'ATM'. The pronunciations of these words are mostly based on English spellings or nativized versions of English spellings. These words are therefore transliterated [7] to better match standard orthography, as described in [8].

3.3. Modeling rare units

Both graphemic and phonemic systems typically contain rare units: letters or phones that are not well represented in the training data. As too few samples lead to poor model estimation, these units are typically removed during system development, often by simply mapping each to its closest acoustic counterpart based on linguistic knowledge. Our approach is similar. In our aim to define a conceptual process that is repeatable across languages, we reduce rare units by merging candidates that both (a) cause minimal phonemic transformation of the training data and (b) do not create an unacceptable number of homonyms. In practice, once G2P mappings are in place, the process consists of the following steps:

- The main phonemic features of each unit are defined based on the linguistic description of the unit.
- Based on their phonemic descriptions, the distance between each pair of phones is measured. (This is a weighted distance, with each type of feature carrying a pre-set weight.)
- Expected phone frequencies are extracted from the transcriptions.
- The merger of any two units is associated with a potential cost, based on a combination of the frequency and distance of changes that result.
- Units whose merger has the lowest cost are considered candidates for merging.
- Of these, any merger candidates that create an unacceptably high number of homonyms are not considered further.
- Merging is continued until the rarest phone is above the necessary occurrence threshold (a value of 30 in the experiments below).

This results in a process whereby – starting from the rarest phones – phones are merged based on closest phonemic distance. Various settings must be determined: the weights used in the phonemic distance measure, the minimum number of occurrences in order to retain a unit and the maximum number of additional homonyms created. Currently these are set based on limited empirical observations, rather than an optimization process. Of course, different levels of granularity will be appropriate given differently sized training sets. We therefore produce different maps (three per language) to match different training conditions. Referred to as 'set1' to 'set3' below, these differ only with regard to the point at which the merger process is halted.

3.4. Syllabification

Once phone strings have been generated for all words, we syllabify these: the aim is to create syllable-like units for sub-word modeling purposes. These units do not have to match actual syllables in the language but should produce 'chunks' of words that can be modeled in a consistent fashion. In addition, we aim to define an algorithm that in itself is language independent, even though it may rely on the language-specific information obtainable from the LSP documents. Our approach is simple:

1. Pre-process and classify vowels and consonants according to the LSP document.
2. Count all consonant clusters occurring at the beginning and end of words in the training data.
3. Syllabify words at consonant clusters VC_1C_2V by creating a syllable boundary between C_1 and C_2.
4. Push the syllable boundary as far to the left of a syllable as possible: try to find the longest valid C_2, such that C_1 is still valid (it has been seen in training); else give preference to longer C_2 and accept invalid C_1; else split before C_1C_2.

This algorithm is well-suited to languages that tend to have open syllables, while its symmetric counterpart (pushing the syllable boundary as far to the right as possible) is more suited to languages that tend to have closed syllables. In Section 4.2 we analyze the ability of this algorithm to mimic true syllables.

4. Analysis

Since the Babel language packs include a manually developed lexicon, we can determine how closely our G2P and syllabification techniques approximate this lexicon. We analyze this in Sections 4.1 and 4.2, before considering STD performance in Section 4.3. Since Telugu was processed in a way that is different from the other languages, we deal with it separately in Section 4.4. For all experiments, we use the latest data packs that were available at the time of analysis[1].

4.1. G2P analysis

The languages studied have quite different G2P characteristics. In Table 3 we summarize, per language, the characteristics of the G2P process when creating the most detailed of the different phoneme sets (set 1). The character count here includes both standard letters and symbols such as hyphen or apostrophe, as these may or may not influence pronunciation, depending on the language itself. As shown in Table 3, most mappings are straightforward one-to-one (1:1) mappings. Only Cebuano requires many-to-many (m:m) mappings: these are caused by rules describing the glottal stop. The large number of Kazakh characters is caused by the use of both Cyrillic and Latin script in Kazakh writing.

When evaluating the mapped lexicons against the official, manually produced Babel lexicons, we consider only those words that occur in both the VLLP transcriptions and the provided lexicons, and evaluate accuracy against both the original lexicon and a remapped version of the reference lexicon (where the set-specific unit mergers are also applied to the reference lexicon). As shown in Table 4 (for set 2, used to develop VLLP systems), languages differ significantly in their predictability. The remapped phone accuracies are all above 80%, but this does not always lead to high word accuracies. The other sets produce very similar results. As to be expected given their definition, set 1 (more units retained) achieves slightly higher accuracy against the original reference, while set 3 (less units retained) achieves slightly higher accuracy against the remapped reference.

Table 3: *The number of characters observed (char), number of mappings applied, and the size of the resulting phoneme set (pho) when creating set 1.*

lang	*char*	*1:1*	*1:m*	*m:1*	*m:m*	*pho*
ceb	29	27	1	9	5	27
kaz	67	65	2	0	0	38
kur	32	31	0	0	0	31
lit	36	30	3	1	0	29
tpi	28	25	1	1	0	23

Table 4: *Word and phone accuracy when comparing the remapped graphemic dictionaries with a mapped and unmapped version of the reference lexicon.*

lang	*map*	*phone acc*		*word acc*	
		orig	*remapped*	*orig*	*remapped*
ceb	set 2	85.7	86.7	48.9	49.7
kaz	set 2	79.7	81.6	36.2	37.2
kur	set 2	98.3	98.9	90.2	93.0
lit	set 2	67.1	82.9	13.4	33.2
tpi	set 2	81.6	92.1	58.5	78.4

4.2. Syllabification analysis

The Babel lexicons cannot be used directly to evaluate the accuracy of the automated syllabification algorithm, as only the pronunciations in the lexicons are syllabified, not the words themselves. Based on the positions of syllable markers in the provided lexicons, syllable markers are therefore inserted into the words themselves, using dynamic programming and a trained scoring matrix [9].

In Table 5 we list the number of unique syllables (*sylls*) generated by two variants of the main syllabification algorithm, as well as the percentage of words that were syllabified correctly (*acc*). A1 is exactly the algorithm described in Section 3.4, while A2 is the same as A1, except that adjoining vowels are also split into syllables: $CVVC$ thus becomes CV and VC.

Table 5: *Automatic syllabification algorithm accuracy measured against the official Babel lexicons.*

lang	*A1 acc*	*A1 # sylls*	*A2 acc*	*A2 # sylls*
ceb	65.54	3 181	70.79	2 392
kaz	71.75	3 668	74.64	3 365
kur	93.57	2 511	94.99	2 409
lit	88.52	4 236	73.25	3 594
tel	86.66	5 380	86.82	5 326
tpi	82.85	2 667	82.93	2 400

Accuracy is high for all languages except Cebuano and Kazakh: the implications of this for STD performance is evaluated next.

4.3. STD analysis

State-of-the-art STD systems were trained, as described in detail in [1]. All systems used multilingual features generated by Brno University of Technology (BUT) [10] and the source text for the language model (LM) comprised both the 3-hour VLLP training text, as well as web text. 'Blind' STD results on the official Babel development set are shown in Table 6, for in-vocabulary (IV), out-of-vocabulary (OOV) and all keywords respectively,

[1] IARPA-babel301b-v2.0b (ceb), IARPA-babel302b-v1.0a (kaz), IARPA-babel205b-v1.0a (kur), IARPA-babel304b-v1.0b (lit), IARPA-babel303b-v1.0a (tel) and IARPA-babel207b-v1.0b (tpi).

using Actual Term-Weighted Value (ATWV) [11] as metric. A further breakdown of results is provided for a system containing only syllables, only whole words, and the combination of these two systems.

While syllable-based results are always worse than the whole-word system for IV words, they are similar or better than whole-word results for OOV words (as also found in [2]). The improvement from system combination is also more significant for OOV words than for IV words. The lower syllabification accuracies obtained for Cebuano and Kazakh do not appear to negatively affect ASR and STD performance, and in all cases, the automated syllables do provide a benefit.

Table 6: *ATWV for syllable-only (S), word-only (W) and word+syllable combination (W+S) VLLP systems when performing STD on the official dev set.*

lang	*IV*			*OOV*			*ALL*		
	S	W	W+S	S	W	W+S	S	W	W+S
ceb	0.289	0.314	0.316	0.252	0.240	0.286	0.282	0.302	0.311
kaz	0.387	0.410	0.422	0.358	0.357	0.399	0.383	0.403	0.419
kur	0.197	0.205	0.206	0.144	0.145	0.177	0.190	0.197	0.203
lit	0.489	0.501	0.513	0.510	0.426	0.524	0.492	0.490	0.514
tel	0.225	0.252	0.263	0.241	0.256	0.279	0.229	0.253	0.267
tpi	0.350	0.373	0.374	0.279	0.291	0.328	0.343	0.365	0.370

4.4. Telugu

Given that the Telugu inherent vowel implies that the order of the orthography does not reflect the order of the acoustic units produced, we investigate different modeling options for Telugu: only utilizing direct mappings (v0); rewriting the orthography using those rules described in the LSP document (v1); and rewriting the orthography using additional rules inferred from the 100 most frequent words (v2). In addition, the three phoneme sets described in Section 4.1 produce three variants of each of the options described above. We first consider the G2P accuracy of the various lexicons, reporting on both variant-based [8] and single-best results (Table 7), before evaluating the implications for ASR and STD.

Table 8 reports the Word Error Rate (WER) and ATWV when using different modeling options. All systems are trained using the VLLP training data and evaluated using the development keywords. In order to analyze the effect of the mappings specifically, a simplified system is used, using only whole word models and not including any additional web data. (The results shown in Table 8 were generated using a LM trained only on the 3-hour VLLP training text, while the system used to generate the results in Table 6 was trained on BUT features that were fine-tuned to the target language, semi-supervised training was employed and the source text for the LM comprised both the 3-hour VLLP training text and web text.) It can be seen that the simplest map results in the best ASR results, but non-negligible gain in STD performance is achieved with the more complex rewrite rules (v0 to v1). However, the large improvement in G2P accuracy from v1 to v2 does not correlate with any improvement in STD results.

5. Conclusion

We have described a range of techniques that can be used to improve ASR and STD when graphemic systems are developed for low-resource languages in a multilingual context. In general, small changes to the graphemic lexicons result in visible STD gain. Interestingly, approximating the true phonemic lexicon more closely does not necessarily predict better STD performance.

Table 7: *G2P accuracy of different Telugu lexicons evaluated against official Babel lexicons.*

rules	*map*	*G2P word acc*		*G2P phone acc*	
		variant	*single*	*variant*	*single*
v1	set2	52.7	58.2	91.8	93.2
v1	set3	53.2	58.2	92.0	93.2
v2	set2	77.9	90.0	96.0	97.9
v2	set3	78.7	90.0	96.2	97.9

Table 8: *ASR and STD dev set results using different Telugu lexicons.*

				ATWV	
rules	*map*	*WER*	*all*	*IV*	*OOV*
v0	set2	78.00	0.1430	0.2095	0.0566
v1	set2	79.00	0.1555	0.2106	0.0840
v2	set2	78.50	0.1522	0.2074	0.0804
v2	set3	78.60	0.1537	0.2099	0.0806

Although our analysis was performed in the context of the Babel project, it is likely that many of these methods will be more widely applicable to speech-technology development for such languages, since they address issues that came up repeatedly for the diverse set of languages that we investigated.

6. Acknowledgment

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of IARPA, DoD/ARL, or the U.S. Government.

7. References

[1] S. Tsakalidis, R. Hsiao, D. Karakos, T. Ng, S. Ranjan, G. Saikumar, L. Zhang, L. Nguyen, R. Schwartz, and J. Makhoul, “The 2013 BBN Vietnamese telephone speech keyword spotting system,” in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 7829–7833.

[2] D. Karakos and R. Schwartz, “Subword and phonetic search for detecting out-of-vocabulary keywords,” in *Proc. INTERSPEECH*, Singapore, Sept. 2014, pp. 2469–2473.

[3] E. G. Schukat-Talamazzini, H. Niemann, W. Eckert, T. Kuhn, and S. Rieck, “Automatic speech recognition without phonemes,” in *Proc. Eurospeech*, 1993, pp. 129–132.

[4] S. Kanthak and H. Ney, “Context-dependent acoustic modeling using graphemes for large vocabulary speech recognition,” in *Proc. Acoustics, Speech and Signal Processing (ICASSP), 2002 IEEE International Conference*

on, 2002, pp. 845–848.

[5] M. Killer, S. Stuker, and T. Schultz, “Grapheme based speech recognition,” in *Proc. Eurospeech*, 2003, pp. 3141–3144.

[6] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Communication*, vol. 56, pp. 85–100, 2014.

[7] W. Basson and M. H. Davel, “Category-based phoneme-to-grapheme transliteration,” in *Proc. INTERSPEECH*, Lyon, France, August 2013, pp. 1956–1960.

[8] M. H. Davel, C. van Heerden, and E. Barnard, “G2P variant prediction techniques for ASR and STD,” in *Proc. INTERSPEECH*, Lyon, France, August 2013, pp. 1831–1835.

[9] M. H. Davel, C. J. van Heerden, and E. Barnard, “Validating Smartphone-collected speech Corpora,” in *Proc. Spoken Languages Technologies for Under-Resourced Languages (SLTU)*, Cape Town, South Africa, May 2012, pp. 68–75.

[10] F. Grézl, M. Karafiát, and K. Vesely, “Adaptation of multilingual stacked bottle-neck neural network structure for new language,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014, pp. 7654–7658.

[11] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, “Results of the 2006 spoken term detection evaluation,” in *Proc. Research and Development in Information Retrieval (SIGIR)*, vol. 7, Amsterdam, July. 2007, pp. 51–57.