UNICODE-BASED GRAPHEMIC SYSTEMS FOR LIMITED RESOURCE LANGUAGES

M.J.F. Gales, K.M. Knill and A. Ragni

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, UK
{mjfg,kate.knill,ar527}@eng.cam.ac.uk

ABSTRACT

Large vocabulary continuous speech recognition systems require a mapping from words, or tokens, into sub-word units to enable robust estimation of acoustic model parameters, and to model words not seen in the training data. The standard approach to achieve this is to manually generate a lexicon where words are mapped into phones, often with attributes associated with each of these phones. Context-dependent acoustic models are then constructed using decision trees where questions are asked based on the phones and phone attributes. For low-resource languages, it may not be practical to manually generate a lexicon. An alternative approach is to use a graphemic lexicon, where the "pronunciation" for a word is defined by the letters forming that word. This paper proposes a simple approach for building graphemic systems for any language written in unicode. The attributes for graphemes are automatically derived using features from the unicode character descriptions. These attributes are then used in decision tree construction. This approach is examined on the IARPA Babel Option Period 2 languages, and a Levantine Arabic CTS task. The described approach achieves comparable, and complementary, performance to phonetic lexicon based approaches.

Index Terms— Low resource speech recognition, graphemic acoustic models.

1. INTRODUCTION

There is a great deal of interest in expanding speech recognition coverage of the world's languages. In many cases the resources available to train large vocabulary continuous speech recognisers are severely limited. The standard phonetically based systems require a lexicon where words are mapped into phones. This is often unavailable or may be inconsistent if derived from multiple sources. Alternatively a grapheme-based speech recognition system [1, 2] could be built. The recogniser then only needs an orthographic lexicon to specify the vocabulary rather than a pronunciation lexicon. For languages with a close grapheme-to-phoneme relation, grapheme based modelling has been shown to be as good as phone based modelling for a wide range of languages including European [1, 2, 3], Arabic [4, 5], African [6, 7, 8], Indian [6] and Asian [9, 10] languages. However, the approaches reported to date have typically used either very limited context questions in decision tree construction, manually crafted them, or automatically derived questions for seen graphemes [1, 2]. This paper proposes a near automatic approach to graphemic speech recognition applicable to lots of languages with different writing systems, with minimal phonetic information and, potentially, very limited acoustic training data i.e. there may be unseen graphemes.

There are four forms of writing system: *Pictographic* - graphemes represent concepts; *Logographic* - words or morphemes; *Syllabaries* - syllables; *Segmental* which can be split into

Alphabet	Consonants and vowels both written
Abugida	Vowels marked as diacritics on consonants
Abjab	Only consonants are marked, vowels optionally written (diacritics)

This work considers graphemic systems for segmental languages.

To date, most graphemic systems have been built either for Latin script languages or the script has been converted to Romanised form before creation of the graphemic lexicon e.g. [9, 10]. The grapheme lexicon is typically created (following some text pre-processing such as lower casing) by modelling each orthographic character as a separate grapheme e.g. `rüstung r ü s t u n g`. Manually derived rules have been used to extend some grapheme sets based on phonetic knowledge and/or position in word [4, 11, 10]. For some languages accented and/or rare characters have been mapped to a common grapheme [4, 12] but at the loss of distinction between these characters. Decision trees (DTs) are used to perform context dependent state tying. Singleton questions about the graphemes to the left and right, and possibly word boundary, in general out perform DTs based on mapping phonetic attributes to the graphemes.

When the amount of audio training data is very limited, some graphemes in a language may be seen rarely or not at all. In addition, the same sound may be represented by multiple graphemes when more than one script is used for a language, such as Kazakh which uses both Latin and Cyrillic scripts. The approaches above do not allow unseen graphemes to be modelled or require phonetic knowledge which may not be available. This paper proposes an automated approach to generating the graphemic lexicon and decision trees that can handle unseen graphemes and complex segmental writing scripts. It exploits the information available in the Unicode Character Database[1] to derive grapheme labels and associated attributes that can be asked as questions in the DTs.

The proposed approach to generating the graphemic lexicon is presented in Section 2. Section 3 and Section 4 describe the data sets and experimental results, respectively. Conclusions are drawn

This work was supported in part by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012 and DARPA under the RATS program via a subcontract to BBN Technologies. The paper does not necessarily reflect the position or the policy of the US Government. 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 expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

[1] http://www.unicode.org

in Section 5.

2. GRAPHEMIC LEXICON

In this work it is assumed that the the text is written in unicode [13]. To illustrate the approach Kazakh, one of the more challenging Babel Option Period 2 languages in terms of its writing system, will be used as an example. Instead of assigning each unicode character in a language's script to a separate grapheme to obtain the grapheme set, the attributes of each unicode character are first obtained. These attributes are then used to map the character into a root grapheme and associated attributes. For example, for Kazakh, which is a mixture of Cyrillic and Latin scripts a subset of the graphemes associated with the letter "I" are

i	G6;D2D3D6	LATIN SMALL LETTER I
I	G6;D8D3D6	LATIN CAPITAL LETTER I
и	G6;D1D2D3	CYRILLIC SMALL LETTER I
ѝ	G6;D1D2D3D4	CYRILLIC SMALL LETTER I WITH GRAVE
й	G6;D1D2D3D5	CYRILLIC SMALL LETTER SHORT I

where the following attributes are defined

D1	CYRILLIC	D2	SMALL	D3	LETTER	D4	WITH GRAVE
D5	SHORT	D6	LATIN	D8	CAPITAL		

All graphemes are thus mapped into a set of core graphemes, and attributes associated with the set of graphemes. This mimics the set of attributes associated with phones that can be obtained for all the e.g. X-SAMPA symbols.

The above scheme has assumed that all unicode characters have a distinct acoustic realisation. Unicode characters that do not have an acoustic realisation or alter the realisation of an adjoining grapheme are split into two distinct groups. The first set are language-dependent graphemes, and are related to diacritics, but written as separate unicode characters, denoted by the word SIGN in the character descriptor. Note VOWEL SIGN characters in for example Abugida written languages are kept as separate symbols with acoustic realisations. For Kazakh there are two such SIGN symbols

ь	D9	CYRILLIC SMALL LETTER SOFT SIGN
ъ	D10	CYRILLIC SMALL LETTER HARD SIGN

These are added as additional attributes to the grapheme on the left of the sign, using the indicator in the second column. The second set of symbols associated with writing schemes. The list below describes these tokens and how they are interpreted [2]

U+0027	APOSTROPHE	add DA to neighbouring grapheme
U+002D	HYPHEN-MINUS	word-boundary
U+005F	LOW LINE	word-boundary
U+200C	ZERO WIDTH NON-JOINER	word-boundary
U+200D	ZERO WIDTH JOINER	no-action

The final information added to the lexicon is word-boundary information, this is also used by default in the CUED phonetic systems and has previously been found to be useful for some languages, including Arabic [5]. The initial grapheme is marked with *^I*, the final grapheme with *^F*, and all others with *^M*. These word boundary attributes have previously been found to be important for Abjab languages such as Modern Standard Arabic (MSA). Thus the Kazakh word for *seven* has the following phonetic (X-SAMPA) and graphemic lexical entries

семь	" s′^I e^M m′^F
семь	G29^I;D1D2D3 G1^M;D1D2D3 G24^F;D1D2D3D9

where the initial '"' in the phonetic pronunciation indicates primary stress, as used in the Babel lexicons. Note for the phonetic system attributes for the phones were obtained from an X-SAMPA look-up table.

For the Babel data, hesitations were marked in the transcriptions using the `<hes>` symbol. For the phonetic systems multiple pronunciations for this token were given for each language. As it is not possible to obtain an explicit graphemic form for these tokens, to address this two `<hes>` specific graphemes were added and used as the pronunciation.

3. DATA

The primary data used for evaluating the graphemic systems were the languages released under the IARPA Babel program [14] for Option Period 2. The list of six languages and official release are given in Table 1.

Language	Id	Release
Kurmanji Kurdish	205	IARPA-babel205b-v1.0a
Tok Pisin	207	IARPA-babel207b-v1.0a
Cebuano	301	IARPA-babel301b-v1.0b
Kazakh	302	IARPA-babel302b-v1.0a
Telugu	303	IARPA-babel303b-v1.0a
Lithuanian	304	IARPA-babel304b-v1.0b

Table 1: Babel Option Period 2 Languages, and data releases

For each of these languages four language packs (LPs) were released, each describing a different subset of data. The sizes of the transcribed sections of the four language packs are (approximate hours of speech and surrounding silence): Full LP (FLP) 40 hours; Limited LP (LLP) 10 hours; Very Limited LP (VLLP) 3 hours; Active-Learning LP (ALP) 1 hour [3]. The data comprises primarily Conversation Telephone Speech (CTS), with a limited amount of scripted data for the the larger LPs (FLP and LLP). For the experiments reported in this work only the transcribed data was used, no semi-supervised training or data augmentation approaches were adopted. The same test data (the supplied language development data) was used for all LPs of a language, this data was not used to tune any of the systems, simply for evaluation. For each language there was approximately 10 hours of CTS distributed. For all the Babel languages, high quality X-SAMPA lexicons are available.

In addition to the Babel data, a Levantine Arabic CTS task was also examined. The data for this task was taken from the RATS program, using the original, not retransmitted, data (channel 0). This yields about 48 hours of transcribed training data. This is similar to the size of a FLP. A 2.5 hour test set was also available. For this Levantine Arabic data no phonetic lexicon was available. However as Arabic has an Abjab written form the attributes of the pronunciation can be manually derived. This will be referred to as the Expert attribute set.

[2] The set of characters listed here are those associated with the Babel languages examined in this paper. More generally punctuation and alphanumeric number mappings must also be dealt with through pre-processing.

[3] The ALP is not meant as a stand-alone set of transcribed data, but to start an active-learning scheme. However here it is used as an example of a very limited resource language.

Language	**System**	**Script**	**Graphemes**†
Kurmanji Kurdish	Alphabet	Latin	62
Tok Pisin	Alphabet	Latin	52
Cebuano	Alphabet	Latin	53
Kazakh	Alphabet	Cyrillic/Latin	126
Telugu	Abugida	Telugu	60
Lithuanian	Alphabet	Latin	62
Levantine Arabic	Abjab	Arabic	36

Table 2: Option Period 2 Languages, and data releases. † the number of graphemes in the FLP, excluding apostrophe.

Table 2 shows some of the attributes of the seven languages investigated. Three different writing schemes were evaluated: Alphabet, Abugida, and Abjab. Four forms of writing script were examined: Latin, Cyrillic, Arabic and Telugu. Additionally the table gives the number of "raw" graphemes, with no mappings, that are observed in the FLP training transcriptions, or the complete Levantine Arabic training transcriptions.

Language Pack	Grapheme Mapping					#
	—	cap	scr	atr	sgn	Phn
FLP	126	67	62	54	52	59
LLP	117	66	61	53	51	59
VLLP	95	59	54	46	44	59
ALP	81	55	51	43	42	59

Table 3: *Number of unique tokens in Kazakh (302) (incrementally) removing:* `cap` *capitalisation;* `scr` *writing alphabet;* `attr` *attributes;* `sgn` *signs*

It is interesting to see how the number of graphemes varies with the form of grapheme mapping used, and the size of the data (or LP). Table 3 shows the statistics for Kazakh, which has the greatest number of observed graphemes as both Cyrillic and Latin script are used. The first point to note is that going from the FLP to the ALP, 45 graphemes are not observed in the ALP compared to the FLP.

As the forms of mapping are increased: removing capitalisation; writing script; remaining grapheme attributes; and sign information, the number of graphemes decreases. However comparing the FLP and ALP, there are still 10 graphemes not seen in the ALP. If the language model is only based on the acoustic data transcriptions this is not an issue. However if additional language model training data is available, then acoustic models are required for these unseen graphemes. In contrast for all LPs all the phones are observed in all LPs. Note for all the phonetic systems, diphthongs are mapped to their individual constituents.

4. EXPERIMENTAL RESULTS

This section contrasts the performance of the proposed unicode-based graphemic systems with phonetic systems, and also an expert derived Levantine Arabic graphemic system. The performance using limited resources on CTS data is poor compared to using larger amounts of resources, or simpler tasks.

4.1. Acoustic and Language Models

The acoustic and language models built on the six Babel languages were built in a Babel BaseLR configuration [14]. Thus no additional information from other languages, or LPs, were used in building the systems. HTK [15] was used for training and test, with MLPs trained using QuickNet [16]. All acoustic models were constructed from a flat-start based on PLP-features, including HLDA and MPE training. The decision trees used to construct the context-dependent models were based on state-specific roots. This enables unseen phones and graphemes to be synthesised and recognised, even if they do not occur in the acoustic model training data [17]. Additionally it allows rarely seen phones and graphemes to be handled without always backing off to monophone models. These baseline acoustic models were then extended to Tandem-SAT systems. Here Bottle-Neck (BN) features were derived using DNNs with PLP plus pitch and probability of voicing (PoV) obtained using the Kaldi toolkit [18] [4]. Context-dependent targets were used. These 26-dimensional BN features were added to the HLDA projected PLP features and pitch features to yield a 71-dimensional feature vector. The baseline models for the Levantine Arabic system were identical to the Babel systems. However the Tandem-SAT system did not include any pitch or PoV features, so the final feature-vector size was 65.

For all systems only the manual transcriptions for the audio training data were used for training the language models. To give an idea of the available data for Kazakh the number of words are: FLP 290.9K; LLP 71.2K; VLLP 25.5K; and ALP 8.8K. For all experiments in this section, manual segmentation of the test data was used. This allows the impact of the quantity of data and lexicon to be assessed without having to consider changes in the segmentation.

4.2. Full Language Pack Systems

Language	ID	System	WER (%)		
			tg	+cn	cnc
Kurmanji Kurdish	205	Phonetic	67.6	65.8	64.1
		Graphemic	67.0	65.3	
Tok Pisin	207	Phonetic	41.8	40.6	39.4
		Graphemic	42.1	41.1	
Cebuano	301	Phonetic	55.5	54.0	52.6
		Graphemic	55.5	54.2	
Kazakh	302	Phonetic	54.9	53.5	51.5
		Graphemic	54.0	52.7	
Telugu	303	Phonetic	70.6	69.1	67.5
		Graphemic	70.9	69.5	
Lithuanian	304	Phonetic	51.5	50.2	48.3
		Graphemic	50.9	49.5	

Table 4: *Babel FLP Tandem-SAT Performance*

To give an idea of relative performance when all available data is used, FLP graphemic and phonetic systems were built for all six Babel languages. The results for these are shown in Table 4. For all languages the graphemic and phonetic systems yield comparable performance. It is clear that some languages, such as Kurmanji Kurdish and Telugu are the hardest, with Tok Pisin (a Creole language) being the easiest. As expected combining the phonetic and graphemic systems together yields consistent performance gains of 1.2% to 1.6% absolute over the best individual systems.

For the Levantine Arabic CTS task no phonetic lexicon was available. However as Arabic uses an Abjab writing form, all conso-

[4] Though performance gains were obtained using FBANK features over PLP, these gains disappeared when pitch features were added in initial experiments.

Tree Questions	WER (%) tg	 +cn
Expert	45.0	43.9
Unicode	45.2	44.2

Table 5: *Levantine Arabic STT performance of graphemic Tandem-SAT systems.*

nants are marked. It is possible to derive attributes from each of these consonants. This was the approach adopted from the graphemic Modern Standard Arabic (MSA) systems in [19] and the system for the RATS data in [20]. This expert derived set of decision tree questions was compared with the unicode-based graphemic approach described in this paper. The results are shown in Table 5. The performance of the two approaches is approximately equal. However it is worth noting that the unicode approach adopted made no use of expert knowledge, simply the attributes of the unicode symbols.

4.3. Restricted Language Pack Systems

One of the aims of this work is to build graphemic systems on very limited acoustic model training data. Again Kazakh is initially considered as it has the largest number of graphemes.

Language Pack	WER (%) phon	 grph	 cnc
FLP	53.5	52.7	51.5
LLP	65.4	64.4	62.9
VLLP	76.2	73.9	73.1
ALP	—	82.0	—

Table 6: *Kazakh WER (%) performance of phonetic and graphemic Tandem-SAT systems.*

Table 6 shows the performance of graphemic systems built on all four of the LPs available. As expected the performance rapidly degrades as the quantity of data decreases. To assess how well the graphemic system was performing, phonetic systems were also built. Note for the VLLP no phonetic information was available. To address this the lexicon, and mispronounced words, were taken from the FLP. For the FLP, LLP and VLLP the graphemic systems outperformed the phonetic ones, with the absolute difference in performance increasing as the quantity of data available decreases. Again performing system combination yielded gains. Note the phonetic ALP system was not built due to the very high error rates obtained.

Language Model	WER (%) phonetic	 graphemic
VLLP	76.2	73.9
FLP	71.2	69.0

Table 7: *Kazakh VLLP acoustic model (CN) performance of phonetic and graphemic Tandem-SAT systems.*

In order to simulate the use of additional language model data, thus including graphemes that were not seen in the acoustic model training data, the FLP language model was used with the VLLP acoustic models. This was done for both the phonetic and graphemic systems. The results are shown in Table 7. For both the phonetic system and the graphemic system, gains of about 5% absolute were obtained from using the FLP LM. What is more interesting is that graphemes not seen in the acoustic model training data, for example "h", exist in the recognition output (and correctly). Thus the proposed approach to generating graphemic systems allows semi-supervised approaches to train models for unseen graphemes.

LP AM	 LM	WER (%) tg	 +cn
FLP	FLP	50.9	49.5
LLP	LLP	61.8	59.5
VLLP	VLLP	71.3	68.6
VLLP	FLP	66.3	63.6

Table 8: *Lithuanian WER (%) performance of graphemic Tandem-SAT systems.*

To examine the performance on a second language, graphemic systems were built on the three larger Lithuanian LPs. Table 8 shows the performance of these systems. The same trends are observed for Lithuanian as Kazakh. For this simpler alphabet, there are 35 distinct graphemes, excluding apostrophe and mapping capitals to lower-case. For the VLLP only two graphemes are not observed ("W" and "X"). Again the FLP LM was used to examine the ability of the graphemic system to handle unseen graphemes. Using the improved FLP LM yielded gains of about 5% absolute. Neither of the unseen graphemes were hypothesised by the system, however they are both very rare ("W" appears in one of the FLP words, "X" four times), and are not in the scoring reference. A similar experiment was also completed on Telugu, where similar to Kazakh, missing graphemes were observed in the recognition output.

5. CONCLUSIONS

This paper has described a simple approach for building graphemic lexicons making use of the unicode character descriptors. In the same fashion as the X-SAMPA phone attributes such as voiced, front, back, can be used for decision tree generation, the unicode character descriptor is converted into attributes that are then used for clustering. By using general attributes combined with state roots of the decision trees it is possible to synthesis unseen graphemes if, for example, additional language model training data is available. The process was applied, with no language-specific knowledge or tuning, to the six Babel Option Period 2 languages: Kurmanji Kurdish, Tok Pisin, Cebuano, Kazakh, Telugu and Lithuanian. Performance of the graphemic and phonetic systems for the FLPs is comparable. For Kazakh it was possible to recognise unseen graphemes, using the VLLP acoustic model with the FLP language model. In addition performance on a CTS Levantine Arabic task was examined. In addition to the languages examined in this paper, the same framework has been applied to Tamil, Assamese, Zulu, Pashto and English. Other than for English, comparable performance was obtained for the graphemic and phonetic systems.

The current implementation has focused on only using the information from the unicode characters. It is possible to combine this information with for example automatically generated (by clustering) questions. This will be investigated in future work.

6. REFERENCES

[1] S.Kanthak and H.Ney, "Context-dependent acoustic modelling using graphemes for large-vocabulary speech recognition," in

Proc. ICASSP, 2002.

[2] T.Schultz M. Killer, S.Stüker, "Grapheme based speech recognition," in *Proc. EUROSPEECH*, 2003.

[3] S. Stüker and T. Schultz, "A Grapheme Based Speech Recognition System for Russian," in *Proc. SPECOM'2004*, 2004.

[4] D. Vergyri et al., "Development of a conversational telephone speech recognizer for Levantine Arabic," in *Proc. INTERSPEECH*, 2005.

[5] Frank Diehl, Mark John Francis Gales, Xunying Liu, Marcus Tomalin, and Philip C. Woodland, "Word boundary modelling and full covariance Gaussians for Arabic speech-to-text systems," in *Proc. INTERSPEECH 2011*, 2011, pp. 777–780.

[6] V.-B. Le et al., "Developing STT and KWS systems using limited language resources," in *Proc. INTERSPEECH 2014*, 2014.

[7] W. Basson and M. Davel, "Comparing grapheme-based and phoneme-based speech recognition for Afrikaans," in *Proc. Pattern Recognition Association of South Africa (PRASA)*, 2012, http://www.prasa.org/proceedings/2012/prasa2012-29.pdf.

[8] F.de Wet R.Molapo, E.Baranard, "Speech data collection in an under-resourced language within a multilingual context," in *4th International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2014.

[9] Viet Bac Le and Laurent Besacier, "Comparison of Acoustic Modeling Techniques for Vietnamese and Khmer ASR," in *Proc. ICSLP*, 2006.

[10] S.Sakti and S.Nakamura, "Recent progress in developing grapheme-based speech recognition for Indonesian ethnic languages: Javanese, Sundanese, Balinese and Bataks," in *4th International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, 2014.

[11] Ramya Rasipuram, Peter Bell, and Mathew Magimai-Doss, "Grapheme and multilingual posterior features for under-resourced speech recognition: a study on Scottish Gaelic," in *Proc. ICASSP*, 2013.

[12] C.Gracia X.Anguera, J.Luque, "Audio-to-text alignment for speech recognition with very limited resources," in *Proc. INTERSPEECH 2014*, 2014.

[13] "The unicode consortium," `httlp://http://www.unicode.org`, Accessed: 2014-09-30.

[14] M. Harper, "IARPA Babel Program," http://www.iarpa.gov/Programs/ia/Babel/babel.html.

[15] S. J. Young et al., *The HTK Book (for HTK version 3.4.1)*, Cambridge University, 2009, `http://htk.eng.cam.ac.uk`.

[16] David Johnson et al., "QuickNet," http://www1.icsi.berkeley.edu/Speech/qn.html.

[17] Kate M. Knill, Mark J. F. Gales, Anton Ragni, and Shakti Rath, "Language Independent and Unsupervised Acoustic Models for Speech Recognition and Keyword Spotting," in *Proc. INTERSPEECH 2014*, 2014.

[18] Daniel Povey et al., "The Kaldi Speech Recognition Toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[19] Frank Diehl, Mark J. F. Gales, Marcus Tomalin, and Philip C. Woodland, "Morphological decomposition in Arabic ASR systems," *Computer Speech & Language*, vol. 26, no. 4, pp. 229–243, 2012.

[20] Mark J. F. Gales and Federico Flego, "Model-based approaches for degraded channel modelling in robust ASR," in *Proc. INTERSPEECH 2012*, 2012.