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. The scheme is applicable to a range of segmental writing systems, requiring no phonetic information. It can also handle limited acoustic training data where 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 are marked as diacritics on consonants
- Abjad 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 ü n g. Manually derived rules have been used to extend some grapheme sets based on phonetic knowledge and/or position in the 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. Questions about the grapheme identity to the left and right, and possibly word boundary, in general outperform 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 based on unicode attributes [13] is presented in Section 2. Section 3 and Section 4 describe the data sets and experimental results.

\(^1\)http://www.unicode.org
respectively. Conclusions are drawn in Section 5.

2. GRAPHEMIC LEXICON

In this work it is assumed that 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

<table>
<thead>
<tr>
<th>D1</th>
<th>CYRILLIC</th>
<th>D2</th>
<th>SMALL</th>
<th>D3</th>
<th>LETTER</th>
<th>D4</th>
<th>WITH GRAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>D5</td>
<td>SHORT</td>
<td>D6</td>
<td>LATIN</td>
<td>D8</td>
<td>CAPITAL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 languages, for example, X-SAMPA phonetic look-up tables.

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, can be 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 skn in the character descriptor. Note that characters in for example Abugida written languages are kept as separate symbols with acoustic realisations. For Kazakh there are two such skn symbols

\[ \text{D9} \quad \text{CYRILLIC SMALL LETTER SOFT SIGN} \]

\[ \text{D10} \quad \text{CYRILLIC SMALL LETTER HARD SIGN} \]

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:

- **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 Abjad 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′1 e′M m′F 
семь  G29′T;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 phone attributes 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.

<table>
<thead>
<tr>
<th>Language</th>
<th>Id</th>
<th>Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kurmanji Kurdish</td>
<td>205</td>
<td>IARPA-babel205b-v1.0a</td>
</tr>
<tr>
<td>Levantine Arabic</td>
<td>204</td>
<td>IARPA-babel204b-v1.0a</td>
</tr>
<tr>
<td>Tok Pisin</td>
<td>207</td>
<td>IARPA-babel207b-v1.0a</td>
</tr>
<tr>
<td>Cebuano</td>
<td>301</td>
<td>IARPA-babel301b-v1.0b</td>
</tr>
<tr>
<td>Kazakh</td>
<td>302</td>
<td>IARPA-babel302b-v1.0a</td>
</tr>
<tr>
<td>Telugu</td>
<td>303</td>
<td>IARPA-babel303b-v1.0a</td>
</tr>
<tr>
<td>Lithuanian</td>
<td>304</td>
<td>IARPA-babel304b-v1.0b</td>
</tr>
</tbody>
</table>

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 comprise 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 were 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, these data were 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 Abjad written form the attributes of the pronunciation can be manually derived. This will be referred to as the Expert attribute set.

3The 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.
Table 2: Attributes of Babel Option Period 2 Languages. † 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 Abjad. 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.

Table 3: Number of unique tokens in Kazakh (302) (incrementally) removing: cap capitalisation; scr writing script; atr 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 all the phones are observed in all LPs. Note if additional language model training data is available, then acoustic models are required for these 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]. Context-dependent targets were used. These 26-dimensional BN features were added to the ILDA 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. Trigram language models were built for all languages. 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. 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, was 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]. Context-dependent targets were used. These 26-dimensional BN features were added to the ILDA 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. Trigram language models were built for all languages. 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

Table 4: Babel FLP Tandem-SAT Performance: Vit Viterbi decoding, CN confusion network (CN) decoding, CNC CN-combination.

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 harder to recognise, 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.

4 Though performance gains were obtained using FBANK features over PLP these gains disappeared when pitch features were added in initial experiments.
Table 5: Levantine Arabic STT performance of graphemic Tandem-SAT systems: \text{Vit} Viterbi decoding, \text{CN} confusion network (CN) decoding.

<table>
<thead>
<tr>
<th>Language Pack</th>
<th>PHON</th>
<th>GRPH</th>
<th>CN CN-decoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLP</td>
<td>53.5</td>
<td>52.7</td>
<td>51.5</td>
</tr>
<tr>
<td>LLP</td>
<td>65.4</td>
<td>64.4</td>
<td>62.9</td>
</tr>
<tr>
<td>VLLP</td>
<td>76.2</td>
<td>73.9</td>
<td>73.1</td>
</tr>
<tr>
<td>ALP</td>
<td>—</td>
<td>82.0</td>
<td>—</td>
</tr>
</tbody>
</table>

Table 6: Kazakh WER (%) CN-decoding performance of phonetic (phon) and graphemic (grph) Tandem-SAT systems: CNC CN-combination.

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

Table 8: Lithuanian WER (%) performance of graphemic Tandem-SAT systems with different Acoustic (AM) and Language (LM) models: \text{Vit} Viterbi-decoding, \text{CN} CN-decoding.

<table>
<thead>
<tr>
<th>Language Model</th>
<th>AM</th>
<th>LM</th>
<th>WER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLP</td>
<td>50.9</td>
<td>49.5</td>
<td></td>
</tr>
<tr>
<td>LLP</td>
<td>61.8</td>
<td>59.5</td>
<td></td>
</tr>
<tr>
<td>VLLP</td>
<td>71.3</td>
<td>68.6</td>
<td></td>
</tr>
<tr>
<td>VLLP FLP</td>
<td>66.3</td>
<td>63.6</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Language Pack</th>
<th>WE (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLP</td>
<td>45.0</td>
</tr>
<tr>
<td>LLP</td>
<td>45.2</td>
</tr>
<tr>
<td>VLLP</td>
<td>43.9</td>
</tr>
<tr>
<td>ALP</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Levantine Arabic STT performance of graphemic Tandem-SAT systems: \text{Vit} Viterbi decoding, \text{CN} confusion network (CN) decoding.

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

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 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 synthesise 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 IARPA 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, similar to the results for European languages in [1].

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


