Multilingual Text-To-Phoneme Mapping

Søren Kamaric Riis, Morten With Pedersen, and Kåre Jean Jensen

Nokia Mobile Phones, Copenhagen, Denmark
soren.riis@nokia.com, morten.w.pedersen@nokia.com, kaare.jensen@nokia.com

Abstract

This paper introduces a novel approach for generating multilingual text-to-phoneme mappings for use in multilingual speech recognition systems. The multilingual mappings are based on the weighted outputs from a neural network text-to-phoneme model, trained on data mixed from several languages. The multilingual mappings used together with a branched grammar decoding scheme is able to capture both inter- and intra-language pronunciation variations which is ideal for multilingual speaker independent speech recognition systems. A significant improvement in overall system performance was obtained for a multilingual speaker independent name dialing task when applying multilingual instead of language dependent text-to-phoneme mapping.

1. Introduction

Speaker independent command word recognition and name dialing on portable devices such as mobile phones and personal digital assistants has attracted significant interest recently. A speech recognition system provides an alternative to keypad input for limited size portable products. The speaker independence makes the system particularly attractive from a user point of view compared to speaker dependent systems. For large vocabularies, user training of a speaker dependent recognizer is likely to become too tedious to be useful.

A phoneme based speaker independent system is ready to use “out-of-the-box” and does not require any training session of the speaker. Furthermore, if the phoneme based recognizer is combined with a text-to-phoneme (TTP) mapping module for generating phoneme pronunciations online from written text, the user may define specific vocabularies as required in e.g. a name dialing application.

Naturally, speaker and vocabulary independence comes at a cost, namely increased complexity for real-time decoding, increased requirements for model storage, and usually also a slight drop in recognition performance compared to speaker dependent systems. Furthermore, speaker independent systems typically contain a number of language dependent modules, e.g. language dependent acoustic phoneme models, TTP modules. For portable devices, the support of several languages may be prohibited by the limited memory available in such devices as separate modules need to be stored for each language.

Recently, systems based on multilingual acoustic phoneme models have emerged [8, 6]. These systems are designed to handle several different languages simultaneously and are based on the observation that many phonemes are shared among different languages [8, 6]. The basic idea in multilingual acoustic modeling is to estimate the parameters of a particular phoneme model using speech data from all supported languages that include this phoneme. Multilingual speech recognition is very attractive as it makes a particular speech recognition application usable by a much wider audience. In addition the logistic needs is reduced when making world wide products. Furthermore, sharing of phoneme models across languages can significantly reduce memory requirements compared to using separate models for each language. Multilingual recognizers are thus very attractive for portable platforms with limited resources.

Even though multilingual acoustic modeling has proven efficient, user definable vocabularies typically still require language dependent TTP modules for each supported language. Prior to running the language dependent TTP module it is furthermore necessary to first identify the language ID of each vocabulary entry.

In this paper, a novel approach denoted multilingual TTP (ML-TTP) for generating pronunciations from written text on-the-fly in a multilingual speech recognition system is presented. The ML-TTP approach removes the need of using a language identification (LID) module and allows for more variation in the pronunciation of vocabulary entries when combined with a weighted branched grammar decoding scheme.

2. Language Dependent TTP Mapping

For applications like speaker independent name dialing on mobile phones the vocabulary entries are typically names in the phonebook that may be changed at any time. Thus, for a multilingual speaker independent name dialer to work with language dependent TTP, a language identification (LID) module is needed. An example of a multilingual speech recognition system using LID is shown in Figure 1. The figure shows how a LID module selects the language dependent TTP module that is used for generating the pronunciation for a recognizer based on multilingual acoustic phoneme models.

Depending on the application, the TTP module may be a statistical model, a rule based model, based on a lookup table that contains all possible words, or any combination of these. The latter approach will typically not be possible for name dialing applications on portable devices with limited memory resources, due to the large number of possible names.

The LID module may be a statistical model predicting the language ID of each entry in the vocabulary, a deterministic module that sets the language ID of each entry based on application specific knowledge, a module that simply requires the user to set the language ID manually or a combination of these. In the most general case, a priori knowledge about the language ID is not available and manual language identification by the user may not be desirable. In that case, language identification must be based on a statistical LID module that predicts the language ID from the written text.

In most applications based on user defined vocabularies, a statistical LID module has very limited text data for deciding the language ID of an entry. For e.g. a short name like “Peter”, only five letters are available for language identification. Furthermore, many names are not unique for a single language but rather used in a large number of languages with different pronunciations. In addition to this, a speaker may pronounce a foreign/non-native name with a significant accent, i.e., the pronunciation of the name is actually a mixture of the pronunciation corresponding to the language from which the name origin-
4.1. TTP mapping module

Four different approaches for TTP mapping have been considered in this work:

- A statistical LID module
- A multilingual TTP module
- A multilingual pronunciation model
- A single TTP module

All these factors imply that the combination of language dependent TTP modules, a statistical LID module and multilingual acoustical phoneme models is likely to give a poor overall performance. Furthermore, if several languages are to be supported in a portable device, the size of the LID and TTP modules may have to be severely limited in order to fit into the low memory resources of the device. For “irregular” languages, like English, high accuracy TTP modules may take up as much as 40–300 kb of memory, whereas TTP modules for rule based “regular” languages like Japanese and Finnish typically require less than 1 kb.

3. Multilingual TTP Mapping

Instead of applying separate TTP modules for each language we propose to use a single TTP module for mapping text directly into pronunciations based on a common multilingual phoneme set. That is, the multilingual TTP module outputs a sequence of multilingual phoneme symbols based on written text as input, see Figure 2. The basic idea is, just as for multilingual acoustic modeling, to observe that a number of phonemes are shared among different languages. Often the same mapping between letters and phonemes exist in several languages, e.g. the letter “p” maps to the phoneme “p” (SAMPA notation) in most contexts for both English, German, Finnish and Spanish. By using the same model for similar mappings, savings in parameters can be obtained compared to using a separate TTP model for each language. Additionally, a multilingual TTP model removes the need for language identification as the text is mapped to a multilingual phoneme string irrespective of language ID.

Naturally, some letters are pronounced quite differently for different languages. The ML-TTP approach will therefore only be successful if the ML-TTP module is capable of producing multiple, alternative phoneme symbols for such letters. The alternative phonemes are then used to create a branched grammar. The principle of branched grammar decoding in combination with a multilingual TTP module is illustrated for the name “Peter” in Figure 3. The name “Peter” is pronounced quite differently in different languages, but using a multilingual TTP module along with branched grammar decoding allows for capturing all pronunciations that are allowed in the set of languages supported by the multilingual recognition system. The different phonemes at each position can be weighted with probabilities as given by the multilingual TTP model, see Figure 3. This is possible when using e.g. neural networks for TTP as they can directly give estimates of the posterior probabilities of the different phonemes for each letter input [4].

It should be noted at this point, that when a Viterbi decoding scheme is used, only one of the possible phonemes are selected at each position during acoustic decoding. However, if an all-path forward decoder is used, all phonemes at a given position gives a weighted contribution to the score of the word, see Figure 3. Thus, a forward decoding scheme will in principle allow pronunciations that are a mixture of several different pronunciations.

4. Experiments

The ML-TTP method for transcribing written text into phonemes has been compared to a number of alternative TTP methods in a multilingual speaker independent name dialing task. Four languages, Finnish, German, English (US and UK), and Spanish, where used for design and evaluation of the overall system. For these four languages the total number of mono-phonemes is 133 corresponding to 39 phonemes for English, 28 for Spanish, 43 for German and 23 for Finnish. However, by defining a common multilingual phoneme set sharing similar phonemes, the total number of mono-phonemes can be reduced to 67 without affecting overall system performance, see [8] for details.

For testing of the overall system recognition rate a Nokia in-house test set for each of the four languages was used. The test set for each language was based on a 120 word vocabulary of names (90 full names, 10 given names and 20 foreign names). The total number of test utterances was 21900: 5038 for German, 5283 for Spanish, 7979 for German and 3600 for Finnish.

A short description of the TTP, LID and multilingual acoustic model architecture, setup, and training is provided below.

4.1. TTP mapping module

Four different approaches for TTP mapping have been considered in this work:
Table 1: TTP training set sizes, model architectures, and model sizes. The number of input, hidden, and output units in the fully connected MLPs are denoted by I, H, and O respectively.

<table>
<thead>
<tr>
<th>Language</th>
<th>No. words</th>
<th>I x H x O</th>
<th>Model size</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK-US</td>
<td>197,277</td>
<td>243 x 104 x 46</td>
<td>30 kb</td>
</tr>
<tr>
<td>German</td>
<td>255,280</td>
<td>217 x 38 x 47</td>
<td>10 kb</td>
</tr>
<tr>
<td>Spanish</td>
<td>36,486</td>
<td>102 x 5 x 32</td>
<td>0.7 kb</td>
</tr>
<tr>
<td>Finnish</td>
<td>15,103</td>
<td>72 x 4 x 25</td>
<td>0.4 kb</td>
</tr>
<tr>
<td>ML</td>
<td>882,915</td>
<td>333 x 99 x 73</td>
<td>40 kb</td>
</tr>
</tbody>
</table>

4.2. LID module

As for the TTP model there are several possible choices for the statistical LID module, e.g. N-grams [1], decision trees [3], and neural networks. In a set of initial experiments, a neural network based LID module was found to have a very good generalization ability for LID classification from short segments of text even for very compact network sizes. Consequently, a standard fully connected feed-forward MLP network was selected for LID classification. The LID neural network takes a symmetrical window of letters as input and gives probabilities for each of the possible languages at the output for the central letter in the window. The overall language probabilities for a given word was computed as a geometrical average over the language probabilities for all letters in the word. The “winning language” for a word was selected as the one with the largest overall probability.

A LID module with four outputs, 10 hidden units and 333 inputs was trained by standard backpropagation on a balanced set of 50,317 words — roughly 12,500 words from each of the four languages. All words where picked at random from the pronunciation lexicons used for training the TTP modules. With 8 bit/parameter precision, this LID module has a size of 10 kb.

On an independent test set of 124,907 Finnish, German, English, and Spanish words the 10 kb LID module gives a classification rate of 86.4% on average.

4.3. Multilingual acoustic module

The acoustic phoneme models where based on a HMM/NN hybrid known as Hidden Neural Networks (HNN) [7]. The basic idea in the HNN architecture is to replace the Gaussian mixtures in each state of an HMM by state-specific MLPs that have a single output and take speech feature vectors as input. A set of 67 multilingual 3-state mono-phoneme HNNs where discriminatively trained on 69,65 Spanish, 9,234 German, 5,611 Finnish, 6,300 US English, and 9,880 UK English utterances taken from the Spanish-VAHA, SpeechDat-AustrianGerman, SpeechDat-Car-Finnish, Timit and WSJCAMO databases, respectively.

The number of hidden units in the network associated with a state in a particular phoneme HNN was selected based on the following heuristics: for phonemes used by a single language, zero hidden units are used. For phonemes shared by two or more languages the number of hidden units is equal to the number of languages sharing the phoneme. With 8 bit/parameter precision this results in a total size of 17 kb for the acoustic models.

Before training, the utterances where mixed with 3 different types of noise (car, café, music) at SNRs in the range 5–20 dB in order to increase noise robustness of the acoustic models.

The noise mixed training files where passed through a standard MFCC preprocessor yielding 13 static, delta and delta-delta coefficients each 10 ms. Each coefficient in the 39 dimensional feature vector was normalized to zero mean and al coefficients corresponding to the log energy were normalized to unit variance.

During decoding of the HNN, a forward decoder was applied. This has been observed to yield a better performance than Viterbi decoding when the acoustic models are trained discriminatively [7]. Furthermore, the all-path forward decoding scheme is more appropriate for branched grammars as described above. Further details about the HNN architecture and training...
Table 2: Word recognition rates for various TTP methods in a multilingual speaker independent name dialing application. A single transcription is used for all entries in the vocabulary.

<table>
<thead>
<tr>
<th>Language</th>
<th>True</th>
<th>noLID</th>
<th>LID</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK English</td>
<td>92.6</td>
<td>87.8</td>
<td>79.4</td>
<td>82.2</td>
</tr>
<tr>
<td>German</td>
<td>95.2</td>
<td>92.2</td>
<td>86.0</td>
<td>87.4</td>
</tr>
<tr>
<td>Spanish</td>
<td>95.4</td>
<td>94.3</td>
<td>91.8</td>
<td>92.3</td>
</tr>
<tr>
<td>Finnish</td>
<td>99.1</td>
<td>98.9</td>
<td>98.5</td>
<td>98.3</td>
</tr>
</tbody>
</table>
Average     | 95.6 | 93.3  | 89.0 | 90.1 |

Table 3: Word recognition rates for various TTP methods in a multilingual speaker independent name dialing application. Branched grammars are used during decoding.

<table>
<thead>
<tr>
<th>Language</th>
<th>True</th>
<th>noLID</th>
<th>LID</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK English</td>
<td>92.6</td>
<td>91.6</td>
<td>81.4</td>
<td>85.8</td>
</tr>
<tr>
<td>German</td>
<td>95.2</td>
<td>93.5</td>
<td>88.7</td>
<td>92.5</td>
</tr>
<tr>
<td>Spanish</td>
<td>95.4</td>
<td>94.5</td>
<td>93.0</td>
<td>96.2</td>
</tr>
<tr>
<td>Finnish</td>
<td>99.1</td>
<td>98.9</td>
<td>98.7</td>
<td>98.8</td>
</tr>
</tbody>
</table>
Average     | 95.6 | 94.6  | 90.5 | 93.3 |

Table 3 illustrates the effect of applying the different statistical TTP modules with a branched grammar decoding scheme. As seen all methods gain a lot from branched grammar decoding and for the Spanish test, the ML-TTP module in combination with branched grammar decoding even outperforms the true TTP transcriptions. This indicates that an ML-TTP model allows for more variation in the pronunciation of words and thereby increases recognition performance. Interestingly, the multilingual TTP model is capable of giving almost the same performance on average for the four languages as the combination of manually set language ID and language dependent TTP (noLID).

Table 4 illustrates the average performance over the four languages when testing in various noise environments with different TTP mapping methods. As can be seen, the gain due to branched grammar decoding and multilingual TTP is maintained in noise.

<table>
<thead>
<tr>
<th>Noise</th>
<th>True</th>
<th>noLID</th>
<th>LID</th>
<th>ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>95.6</td>
<td>94.6</td>
<td>90.5</td>
<td>93.2</td>
</tr>
<tr>
<td>Volvo80</td>
<td>92.3</td>
<td>90.6</td>
<td>85.6</td>
<td>88.6</td>
</tr>
<tr>
<td>Babble</td>
<td>87.6</td>
<td>84.8</td>
<td>80.0</td>
<td>82.6</td>
</tr>
<tr>
<td>Pop Music</td>
<td>83.2</td>
<td>79.8</td>
<td>74.5</td>
<td>77.0</td>
</tr>
</tbody>
</table>

5. Results

The results presented below gives the overall speech recognition performance when using the TTP methods described above for transcribing text in combination with the multilingual HNN speech recognizer.

Table 2 shows the performance when using the different TTP methods for each of the four test languages when a single pronunciation is used for each entry in the vocabulary (no branched grammar decoding). The column “TrueTTP” shows the performance obtained using hand-transcriptions, “noLID”, the performance obtained assuming known language ID of each entry and language specific TTP modules, and “LID” the performance obtained when using a LID module in combination with language dependent TTP modules. The last column in the table shows the performance of a system employing a multilingual TTP module.

As seen from the table, the true transcriptions are clearly superior for all languages. However, for applications intended for portable devices, there is usually not room for storing a complete dictionary so the TTP mappings must be based on a more compact statistical method. Comparing the rows entitled “noLID” and “LID” it is evident that the errors in language ID introduced by the statistical LID module seriously hampers the recognition performance. Thus, even with a LID module that gives a fairly accurate language identification, the performance of the overall system is seriously affected by incorrect language identification for a few words.

Table 3 illustrates the performance obtained assuming known language ID of each entry and language dependent TTP modules. As can be seen, the gain due to branched grammar decoding and multilingual TTP is maintained in noise.

6. Conclusion

We have introduced a novel approach for generating multilingual text-to-phoneme mappings for use in multilingual speech recognition systems. The approach is based on the ability of the statistical TTP module to generate a weighting of the phonemes at each position. Here we have used the phoneme posterior probabilities provided by a feed-forward neural network, trained on data mixed from several languages.

The set of weighted phonemes at each position are used to create a branched grammar, using all-path forward decoding. The branched grammar allows for capturing of both inter- and intra-language pronunciation variations.

Tests showed that the multi-lingual TTP approach along with the branched grammar offers a significant improvement in recognition performance in multi-lingual phoneme based speaker independent speech recognition compared to using language dependent TTP modules in combination with a LID module. In some cases the multi-lingual TTP approach even improved recognition performance compared to when the language ID was known a priori.

7. References