Text-To-Speech with cross-lingual Neural Network-based grapheme-to-phoneme models

Xavi Gonzalvo, Monika Podsiadło

Google

{xavigonzalvo, mpodsiadlo}@google.com

10.21437/Interspeech.2014-177

Abstract

Modern Text-To-Speech (TTS) systems need to increasingly deal with multilingual input. Navigation, social and news are all domains with a large proportion of foreign words. However, when typical monolingual TTS voices are used, the synthesis quality on such input is markedly lower. This is because traditional TTS derives pronunciations from a lexicon or a Grapheme-To-Phoneme (G2P) model which was built using a pre-defined sound inventory and a phonotactic grammar for one language only. G2P models perform poorly on foreign words, while manual lexicon development is labour-intensive, expensive and requires extra storage. Furthermore, large phoneme inventories and phonotactic grammars contribute to data sparsity in unit selection systems. We present an automatic system for deriving pronunciations for foreign words that utilises the monolingual voice design and can rapidly scale to many languages. The proposed system, based on a neural network cross-lingual G2P model, does not increase the size of the voice database, doesn't require large data annotation efforts, is designed not to increase data sparsity in the voice, and can be sized to suit embedded applications.

1. Introduction

We aim to improve the performance of a text-to-speech system on out-of-vocabulary foreign words. High-quality speech synthesis is expected to be robust against such words even if they do not conform to the orthographic or pronunciation rules of the language of the synthesiser. Typical scenarios are synthesising social network contact names, navigation directions abroad, and many others. These scenarios are very frequent in TTS usage and developers can no longer afford to ignore them.

TTS systems rely on a lexicon containing as many words as possible with their associated pronunciations. The pronunciations utilise a monolingual sound inventory and are composed according to a monolingual phonotactic grammar that constrains the distribution of phonemes in a given language. G2P models can be trained on such a lexicon to provide pronunciations for out-of-lexicon words. Such a design optimises for performance on single-language input. The voice contains a rich inventory of units representing the phoneme distribution of that language, and G2P models work pretty well on words that follow similar rules as the training data. However, when words not conforming to such rules are encountered, TTS systems still try to apply native rules to guess the pronunciation. Currently, existing solutions rely on simply adding foreign words to the original lexicon. This requires manual effort and is a solution only applicable to server-side systems without space limitations. In embedded systems lexicons are heavily pruned and foreign words are typically first to be excluded. In addition, extending the lexicon is not a robust solution; it neither scales well nor is generalisable enough. Moreover, foreign words not explicitly marked as such do not contribute to the G2P model as they often pollute it.

A better approach is to define phoneme mapping tables between two different languages. This has been attempted both in the TTS domain [1] where a phonetic similarity function was proposed, and in the ASR domain [2]. An example of phoneme mapping was presented in [3] for German and English. A similar system was discussed in [4, 5] where foreign speech waveforms were selected based on the acoustic similarity to a native voice. Nativization in TTS systems has been addressed by analogy [6]. Similarly, interpolation of G2P models has been proven useful in dealing with accents [7].

Phoneme mapping tables can be extracted manually or by automatic means [1]. In this approach, phonemes of the foreign voice must be replaced by the most similar sounds available in the voice acoustic database. Authors proposed a general language-independent algorithm intended to convert phonemes from one language to the other. The algorithm is based on a similarity function using phonetic-articulatory features.

One important consideration is to what extent nativised pronunciations will match user expectations and consequently improve intelligibility. In some cases it has been shown that foreign accent in strongly assimilated words or short inclusions actually improves acceptability [8]. We hypothesise that nativising pronunciations will have an overall positive effect, especially for more familiar words.

This paper proposes an extension to TTS systems to deal with pronunciation of foreign words. We propose a method to automatically generate nativised pronunciations for any combination of native and foreign language with an application to TTS systems and a cross-lingual G2P system based on neural networks (NN). Firstly, cross-lingual lexicons are built using finite state transducers (FST) representing the phoneme mapping. This process is semi-automatic since some fine tuning may be necessary. The second step involves the training of a cross-lingual G2P system. The system as a whole can be defined as being embeddable, cross-lingual and easily extendable to other languages. The terms "Multilingual" and "cross-lingual" will be used in different contexts in the rest of the paper. We refer to "multilingual" as a system than can deal with multiple languages. "Cross-lingual" is used to designate how the TTS handles multiple languages internally, that is, using a single phoneme inventory and nativised pronunciations.

This paper is organised as follows. Section 2 describes the system and the phoneme mapping strategy. Section 3 defines the process to create cross-lingual lexicons. Section 4 describes the G2P system. Section 5 presents two types of experiments, objective and subjective. Finally, Section 6 concludes the paper.

2. System overview

The idea is to develop a multilingual TTS system using a single G2P model trained with M languages and a single phoneme inventory. The proposed solution consists of two steps described below (see Figure 1).

First a mapping table is built in the linguistic space. This table usually contains mappings which are possibly ambiguous (ie., linguistic features cannot differentiate two phonemes). Disambiguation is then performed in the acoustic space. The acoustic cross-lingual mapping is used to reduce the number of mapping cases obtained in the phonetic space. This employs the acoustic distance between states of the Hidden Markov Models (HMMs) of the two different languages.

Second, the set of foreign languages ($L_i^{(f)}, i \in [1, M]$) and their corresponding phoneme mapping from a new cross-lingual lexicon ($L^{(c)} = [L_1^{(mf)} \dots L_M^{(mf)}]$). This lexicon is built using the phoneme inventory of the native language $L^{(n)}$ and the mapped pronunciations from each foreign language ($L_i^{(mf)}$, $i \in [1, M]$).

Finally, a single NN is trained with the cross-lingual lexicon containing M languages. At runtime, the user specifies the native language (ie. the main language of the TTS). Foreign words can then be pronounced in a different language using the cross-lingual G2P.

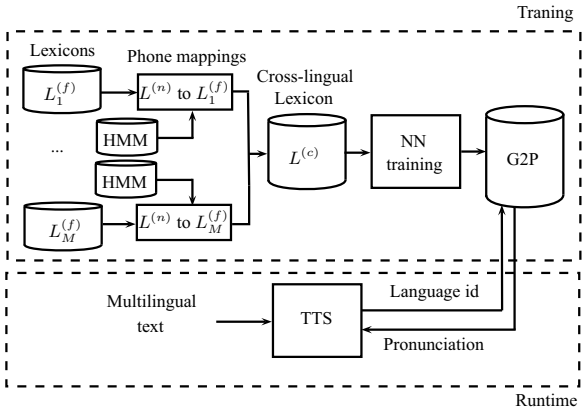

Figure 1: *System Workflow. There are two parts, training and runtime. During training, a cross-lingual lexicon is built using M mappings. Second, this cross-lingual lexicon is used to train the NN G2P. During runtime the user can specify the language identification to select how each word should be pronounced.*

3. Cross-lingual lexicons

Nativisation of lexicons is performed using phoneme mapping tables. These tables are constructed offline and can include some context. This gives the system some flexibility when there are non-existent sounds in a language (eg. sound /sh/ in Spanish). Stress is mapped as well and it is transplanted at the syllable level. The phoneme mapping table is automatically converted into a Thrax grammar [9] and that into an FST so that the decoding process involves a composition step with the input pronunciation. A reduced version of the phoneme mapping transducer for Spanish speaking English is shown in Figure 2.

A similar mapping table in the feature space was first proposed by [5]. In this work, context-independent phone mappings derived from articulatory-feature-space distance measures

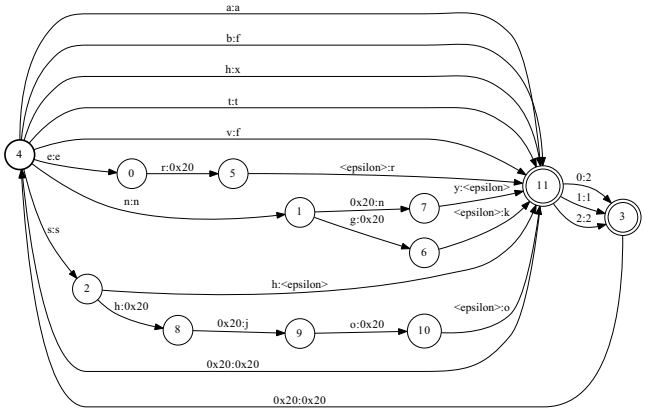

Figure 2: *Spanish speaking English phoneme mapping represented as an FST. Some trivial mappings have been deleted to simplify transducer.*

are used. Another interesting example can be found in [8] and also used in [10]. The idea is to represent each phoneme as a vector of articulatory features, according to the concepts of classical phonetics. Here we are using a combination of three techniques to build the tables: phonetic similarities, acoustic mapping and human intervention if necessary.

Our initial phoneme mapping is based on the assumption that two phonemes are perceived as similar when they have similar phonetic-articulatory features [1]. The result is a phonetic similarity function that gives a quite general classification of phonemes as defined by articulatory phonetics.

The distance between two phonemes, x_n and x_m is:

$$D(x_n, x_m) = \sum_{i=1}^{F} d_i(x_n, x_m) \tag{1}$$

where $d_i(x, y)$ is the distance function for feature i between phonemes x and y defined as:

$$d_i(x, y) = \begin{cases} 1, & \text{if } g(x, i) = g(y, i) \\ 0, & \text{if } g(x, i) \neq g(y, i) \end{cases} \tag{2}$$

being F the number of features, $g(x, i) \in [0, 1]$ the value of feature i for phoneme x.

Note that Equation 1 could include a weight for each feature as in [1] where the importance of each feature is set iteratively to find the optimal weights in order to find the best mapping among all phonemes. In our case this is not necessary because the disambiguation takes place in the acoustic domain. We use a total of 35 features: (1) flags: vowel, voiced, rounded; (2) length (ie. short, long, diphthong), height (eg. close), position (eg. front), place of articulation (eg., dental) and manner (eg., nasal).

The final disambiguation between phoneme mappings is done taking into account the phonetic and acoustic distance. We define the latter as the distance between the states of two sets of HMMs, one for each language (ie. native and foreign). Euclidean distance between the means of the mixture of Gaussians of the central state is used to measure the cost of matching two HMM states.

HMMs (3 states left-to-right no skip) with 10 Gaussian mixtures are trained for each phoneme. By finding the correspondence of each Gaussian mixture model of each state between the HMMs of both languages we can get an estimate of

how well two phonemes map with each other. An unsupervised clustering algorithm [11] is used to map Gaussians of the two language models of the central state only. As each language is speaker dependent, the unsupervised clustering is performed by iteratively mapping Gaussians and updating a linear transformation between models [12].

4. Grapheme-to-phoneme conversion

There are multiple approaches to G2P. Initial approaches used graphs and more advanced techniques used joint-sequence models [13]. Other approaches presented WFST and EM training [14]. All these techniques have been demonstrated to perform very well. NNs have also been widely applied in the field since [15]. A common solution is to use lots of neurons with one single layer. Other studies even discuss the idea of not using a prior-alignment [16]. [17] proposes a method to automate the data preparation for the training of a neural network performing multilingual G2P conversion. A recent approach using long-short term memory (LSTM) was presented in [18]. The main reasons to use NNs are: (1) Footprint (size of weight matrices can use a floating point implementation); (2) Embeddable (memory consumption at runtime is very small); (3) Fast decoding; (4) Multilingual. Multiple languages are trained in the same model.

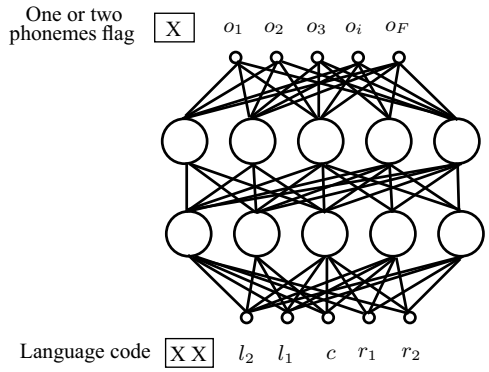

Figure 3: *NN used for cross-lingual G2P. Language code is a binary representation of the input language. Input grapheme is encoded as a one-of K graphemes. Output is encoded as a set of phonetic features.*

What we propose is to use a NN where multiple languages are trained simultaneously using a single phoneme inventory. The neural network is shown if Figure 3 where the input is a grapheme with left and right context and a language identifier. The output is a set of phonetic features that we decode using a L1 distance to obtain the closest estimate.

NNs with multiple hidden layers can represent some functions more efficiently than those with one hidden layer. The challenge, though is the amount of data and the performance of the training process. Nevertheless, deep NNs have gained a lot of attention during last years, especially for Automatic Speech Recognition [19] and TTS [20] applications. This is due to the recent progress in hardware and software. As it has been shown for ASR, neural networks are adequate for multilingual system [21].

The process to use the proposed model consists of three steps: dictionary alignment, the neural network training and the decoding.

4.1. Dictionary Alignment

This step aims to align grapheme and phoneme tokens in a pronunciation dictionary applying many-to-many mapping using the WFST paradigm [14] and forward-backward training. The forward part uses a Expectation-Maximisation approach. The expectation step collects partial counts for each grapheme-phoneme-sequence pair and then a maximisation-step function simply normalises the partial counts to create a probability distribution. The backward part generates the sequence pair. Once the probabilities are learnt, the Viterbi algorithm can be used to produce the most likely alignment.

4.2. Model building

In this step a model is trained using the aligned dictionary to map graphemes to phonemes. The model is represented as a fully connected feed-forward NN architecture similar to the original NETtalk model [15] with one block for K input graphemes (each representing one letter of the alphabet as a binary code), one block of $1 + 2F$ output features, where F is the number of phonetic features representing one phoneme, and one flag indicating if the output involves two phonemes. Graphemes corresponding to the shortened string of phonemes are accommodated by allowing a null phoneme. Each input word starts off with its first grapheme aligned with the middle input block, and then moves to the left, one grapheme at a time, until its last grapheme is aligned in the centre. The network is trained so that the output phonemes correspond to the grapheme represented by the central input block.

With the pre-aligned training data we can use a standard learning algorithm such as the back-propagation algorithm. There are a number of possibilities for cross-lingual mapping. One possibility would be to train multiple models [22], one for each language. We have opted for a single model by design assuming that the word accuracy of the final G2P model will not be affected.

4.3. Decoding

In the decoding step we aim to generate pronunciations for novel and known words. The output includes articulatory features, stress encoding and a flag to indicate whether the current grapheme maps to one or two phonemes.

As the output feature vector may not yield a perfect match with the desired vectors of any of the phonemes, a best match procedure is used to choose the phoneme from the features. This procedure chooses the best phoneme $\hat{x}$ whose feature vector is closest to the network's output vector, that is,

$$\hat{x} = \underset{x_j \in [1,S]}{\arg\min} \sum_{i=1}^{F} |o_i - g(x_j, i)| \quad (3)$$

where x_j is one of the S phonemes, F is the total number of features in the output of the network, $o_i \in [0, 1]$ is the output of the network for feature i and $g(x_j, i) \in [0, 1]$ is the value of feature i for phoneme x_j which is used in Equation 2.

5. Experiments

We have conducted a series of objective and subjective evaluation tests. On the one hand, the objective experiments show the word and phoneme accuracy for different configurations and helps us choose the best one. On the other hand, the subjective test aims to evaluate the quality of the synthesiser when the proposed cross-lingual module is used to pronounce foreign words.

5.1. Objective evaluation

In this experiment we have tested our cross-lingual G2P mapping system with different configurations. First we did an experiment with en-US only where we compared the results with a state-of-the-art WFST system versus a NN. The size of the total lexicon is 340k words and we have used 70% for training and 30% for testing.

The weights of the NN were initialised randomly with ± 0.3, then optimised to minimise the mean squared error between the output features of the training data and predicted values using a GPU implementation of a mini-batch stochastic gradient descent (SGD)-based back-propagation algorithm. NN was trained using a step size of 0.1, momentum of 0.9 and a Sigmoid activation function was used for both hidden and output layers. Results for US English can be seen in Table 1. The network has two hidden layers and 200 neurons which is the optimum configuration that guarantees good performance and footprint. The input and output layers' dimensions are 365 and 71, respectively.

Table 1: *Accuracies (in percentage) of NN-based G2P system with en-US lexicon. Size of the model is in kbytes. The context for WFST system is described in terms of regular n-grams. For the NN system, the context is left and right so it includes the number of graphemes apart from the central one.*

System	Context	Size	Word	Phoneme	Stress
WFST	3-gram	350	45.76	84.25	-
WFST	5-gram	2200	**56.21**	86.88	-
NN	4	340	47.54	80.68	87.08
NN	6	430	**55.26**	82.83	89.75

As shown above, WFST using fivegrams is the best system. It is also the most expensive in terms of size. The most balanced configuration is NN using 6 contexts where word accuracy is about 10% better than WFST using trigrams while the footprint of the model is still very small.

In the second set of experiments we have trained a cross-lingual NN with British English speaking 4 languages (ie. Spanish, French, Italian and German). We have opted for the British lexicon as the word accuracy is higher. The total size of the lexicons is about 1.2M words where 70% is used for training and 30% for testing, maintaining the percentage for each languages. Results using multiple configurations can be seen in Figure 4.

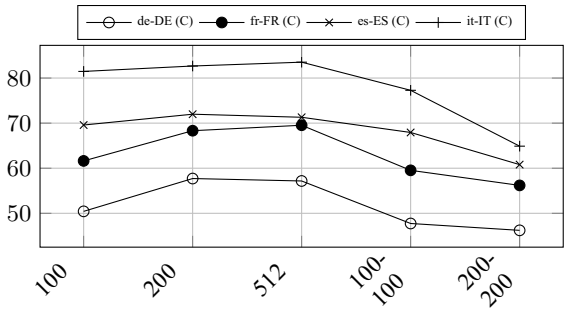

Figure 4: *Word accuracy in percentage for different hidden layer configurations.*

In this case, a single layer with 200 neurons is the best configuration. Results comparing the system trained with one language or using the cross-lingual lexicon can be seen in Table 2. This table also shows the relative contribution of each language to the overall model. This is important since the accuracy for each language is directly related to the size of the original monolingual lexicon and its complexity as a language (eg. number of vowels). These results demonstrate that training one single NN with multiple cross-lingual languages affects the quality only by a percentage that is about less than 5% in most cases.

Table 2: *Word accuracy (in percentage) of NN-based G2P system with en-GB lexicon speaking 4 foreign languages.*

Language	Cross-lingual	Monolingual	Size (%)
Spanish	71.27	77.01	17.84
French	67.81	69.0	35.19
Italian	82.1	84.2	30.85
German	56.55	64.47	16.11

5.2. Subjective evaluation

Subjective evaluation was conducted in the form of AB tests. A test set containing 147 foreign test items was constructed. Test words were various location names (e.g., cities), famous landmarks (e.g., Buckingham), personal names (eg., Michael), popular proper names (e.g., Hangouts). Each AB test pair was evaluated 10 times using naive native-speaking raters. Results are in Table 3 where you can see a set of native languages (ie. Spanish, German, Italian and French) synthesising British English input.

As you can see, our system performed significantly better than the baseline, having 52% preference rate for the cross-lingual system when averaged over all languages. The TTS system is an HMM-based speech synthesis trained with 24-th order LSP and a simple mixed excitation vocoder [23].

Table 3: *AB comparison test for a native language speaking British English.*

Language	Cross-lingual	Monolingual	No preference
Spanish	**44.1**%	20.2%	35.7%
German	**44.5**%	25.5%	30%
Italian	**56**%	18.3%	25.7%
French	**56.0**%	15.6%	28.4%

6. Conclusions

We have presented a multilingual system that generates nativised pronunciations for foreign words while respecting the monolingual voice design. We have shown how cross-lingual lexicons can be built and a NN-based G2P system trained for all languages in a single model. We have shown that this approach works well for improving the quality of a TTS on out-of-vocabulary foreign words. Lastly, we have shown that the G2P system based on NN is a good model to train multiple languages simultaneously maintaining high word and phoneme accuracy while reducing the total footprint of the model compared to a state-of-the-art WFST.

Our approach has been proven to work well with Indo-European languages such as English speaking French. We are now working to adapt it to Asian languages like Cantonese and Japanese. Here, with the importance of tones or pitch accents, mapping in prosodic space might be necessary, similar to what has been reported in [5].

7. References

[1] L. Badino, C. Barolo, and S. Quazza, "A general approach to TTS reading of mixed-language texts," in *INTERSPEECH*, 2004.

[2] K. C. Sim and H. Li, "Robust phone set mapping using decision tree clustering for cross-lingual phone recognition," in *ICASSP*, 2008.

[3] P. Olaszi, T. Burrows, and K. Knill, "Investigating prosodic modifications for polyglot text-to-speech synthesis," in *MULTILING2006*, 2006.

[4] N. Campbell, "Foreign-language speech synthesis," in *ESCA ETRW on Speech Synthesis*, 1998.

[5] ——, "TALKING FOREIGN - concatenative speech synthesis and the language barrier," in *INTERSPEECH*, 2001, pp. 337–340.

[6] T. Polykova and A. Bonafonte, "Introducing Nativization to Spanish TTS Systems," in *Speech Communication*, 2011.

[7] T. Li, P. C. Woodland, F. Diehl, and M. J. F. Gales, "Graphone model interpolation and arabic pronunciation generation," in *INTERSPEECH*, 2011.

[8] B. Pfister and H. Romsdorfer, "Mixed-Lingual Text Analysis for Polyglot TTS Synthesis," in *EUROSPEECH*, 2003, pp. 2037–2040.

[9] W. S. T. Tai, R. Sproat, "Thrax: An Open Source Grammar Compiler Built on OpenFst," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2011.

[10] L. Tomokiyo, A. Black, and K. Lenzo, "Foreign accents in synthetic speech: development and evaluation," in *INTERSPEECH*, 2005, pp. 1469–1472.

[11] G. F. K. Rose, E. Gurewitz, "A deterministic annealing approach to clustering," in *Pattern Recognition letters 11*, 1990, pp. 589–594.

[12] Y. Agiomyrgiannakis and X. Gonzalvo, "Methods and Systems for Automated Generation of Nativized Multi-Lingual Lexicons," Patent US 14/053,052, 10 14, 2013.

[13] Bisani and Ney, "Joint Sequence models for Grapheme-to-Phoneme Conversion," in *CSL*, 2008.

[14] Jiampojamarn, G. Kondrak, and T. Sherif, "Applying Many-to-Many Alignments and Hidden Markov Models to Letter-to-Phoneme Conversion," in *NAACL HLT*, 2007, pp. 372–379.

[15] T. J. Sejnowski and C. R. Rosenberg, "Nettalk: a parallel network that learns to read aloud," in *Neurocomputing - IJON 01*, 1986.

[16] J. Bullinaria, " Text to Phoneme Alignment and Mapping for Speech Technology: A Neural Networks Approach," in *IJCNN*, 2011, pp. 625–632.

[17] H. Hain, "Automation of the training procedures for neural networks performing multi-lingual grapheme to phoneme conversion," in *EUROSPEECH*. ISCA, 1999.

[18] F. B. H. Sak, A. Senior, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," in *Neural and Evolutionary Computing*, 2014.

[19] A. Senior and X. Lei, "Fine Context, Low-rank, Softplus Deep Neural Networks for Mobile Speech Recognition," in *ICASSP*, 2014.

[20] H. Zen and A. Senior, "Deep Mixture Density Networks for Acoustic Modeling in Statistical Parametric Speech Synthesis," in *ICASSP*, 2014.

[21] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of Deep Neural Networks," in *ICASSP*, 2013.

[22] E. Beatrice, "Text-to-Phoneme Mapping Using Neural Network," in *PhD*, 2008.

[23] X. Gonzalvo, J. Socoro, I. Iriondo, C. Monzo, and E. Martinez, "Linguistic and Mixed Excitation Improvements on a HMM-based speech synthesis for Castilian Spanish," in *SSW6*, 2007.