Letter-To-Phoneme Conversion based on Two-Stage Neural Network focusing on Letter and Phoneme Contexts

Kheang Seng¹, Yurie Iribe², Tsuneo Nitta¹

¹Graduate School of Computer Science and Engineering, Toyohashi University of Technology, JP ²Information and Media Center, Toyohashi University of Technology, Japan

kheang@vox.tutkie.tut.ac.jp, iribe@imc.tut.ac.jp, nitta@cs.tut.ac.jp

Abstract

The improvement of Letter-To-Phoneme (L2P) conversion that can output the phoneme strings corresponding to Out-Of-Vocabulary (OOV) words, especially in English language, has become one of the most important issues in Text-To-Speech (TTS) research. In this paper, we propose a Two-Stage Neural Network (NN) based approach to solve the problem of conflicting output at a phonemic level. Both Letter and Phoneme Context-Dependent models are combined and implemented in the first-stage NN to convert several letters into several phonemes. Then, the second-stage NN can predict the final output phoneme by observing on a combination of several consecutive phoneme sequences that obtained from the first-stage NN. Therefore, our L2P conversion module takes a sequence of letters as input and outputs only one phoneme at each time. By focusing mainly on the result of word accuracy of OOV words, this new approach usually provides a higher performance.

Index Terms: Letter Context-Dependent model, Phoneme Context-Dependent model, Phoneme Sequences Pattern-Observation model, Many-to-Many alignment and Two-Stage Neural Network based approach.

1. Introduction

In general, the hardness of TTS system development depends firstly on the complexity of pronunciation rules in each language: English pronunciation has been noticeably the most difficult case since it has been influenced by many foreign languages around the world. Theoretically, TTS system has been divided into several modules [1], [2] including the L2P conversion module.

For improving the predictive ability of L2P conversion system to generate the new phoneme string from any of OOV words, many approaches have been proposed. The majority of those approaches mentioned that the presence of Letter Context-Dependent (LCD) model allows increasing effectiveness of the phoneme as well as word accuracy of L2P conversion system. On the other hand, some other approaches also showed the usefulness and effectiveness of Phoneme Context-Dependent (PCD) model [5].

Concerning to the problem of conflicting output at the phonemic level as shown in Figure 1, it is very difficult to say which phoneme (i.e. /IH/ or /AH/) corresponds to the letter ‘l’ or letter sequence “ACTICAL”. There is no any approaches in which discuss how to solve it yet. Therefore, this paper proposes a new idea to improve the L2P conversion module by using the two-stage NN based approach as main model and focusing on both letter and phoneme context information. In addition to this new approach, it can solve the mentioned problem as well.

\[ \text{Slicing Window} = 7 \text{ letters} \rightarrow 1 \text{ corresponding phoneme} \]

\[ \text{PR ACT ICAL ITY} \rightarrow \text{/PR AEKT IH KAE LAHTIY/} \]

\[ \text{IMPR ACT ICAL} \rightarrow \text{/IMPR AEKT AH KAH1/} \]

Figure 1: Conflicted output phonemes of the letter ‘l’ on sequence “ACTICAL”

The remainder of this paper will describe firstly some other interesting approaches for L2P conversion in section 2. Section 3 explains how to prepare an aligned dictionary to be used along our experiments by applying a Many-to-Many alignment method available in GIZA++ open source toolkit. Section 4 presents the most important techniques of our global work for getting higher accuracy, while different results are reported in section 5. Section 6 discusses our current approach and provides some ideas for further improvement of the L2P conversion module. Finally, section 7 concludes this paper.

2. Related work

In response to the lack of portability and predictive ability of L2P conversion, many interesting data-driven approaches have been proposed and widely used with many interests.

Pronunciation-by-Analogy [3], known as PBA approach and proposed in 2000 by Marchand and Damper. They reported 65.5% of word accuracy on NETTalk dataset.

The staged back-propagation neural network approach [6] provided 82% as word accuracy, while they trained it with 7000 most common words in American English and tested with 10000 words including the previous training dataset.

By supposing the L2P conversion problem as a statistical machine translation problem [8], they reported 63.81% of word accuracy based on an aligned_CUDict dataset [9].

Another approach based on an inference of rewriting rules [7], which follows the idea of Decision Tree algorithm, was created as a tool called “IrisaPhon”. Many tests were done by investigating with different languages or datasets. For the same aligned_CUDict dataset, 74.40% was reported as word accuracy by measuring in term of word precision averaged on the whole dataset.

In addition, Hidden Markov Models (HMM) based on the Context-Sensitive model [4] and HMM with Context-Sensitive Observations model [5] were proposed by Taylor P. in 2005 and Kalu U. O. et al. in 2010, respectively. The second HMM approach showed that the PCD model can also provide a good result after conducting an experiment with Unilex (UK) dataset, but not with aligned_CUDict (US) dataset that contains many loan words as well as errors.

¹ In an aligned dictionary, the length of each letter sequence (word) and its corresponding phoneme sequence must be equal to each other.
3. Letter-to-Phoneme alignment

Concerning to some previous researches that studied on the L2P conversion problem, it was emphasized that the quality of aligned dictionary (dataset) can affect badly to the accuracy of L2P conversion system [10]. Therefore, the dictionary with high consistency, which ensures the association between each letter and the corresponding phonemes, is always required to improve the performance of L2P conversion module [5], [6].

After analyzing the aligned CMUDict [9] which has been used by many researchers, many errors were found as shown in Figure 2 and also on the second column of Table 1.

By the way, the GIZA++ toolkit provides especially a many-to-many alignment algorithm that can map several letters to several phonemes and vice versa. In this paper, instead of using GIZA++ for modeling the whole process of L2P conversion module [8], we just used it as a preprocessing step to reproduce the new aligned dictionary with high consistency for our experiments.

The process of regenerating the new aligned dictionary is described in Figure 3. First, the unaligned dictionary that contains many pairs of word and corresponding phoneme string, it is trained with GIZA++ toolkit. Next, a new file written under a specific format of GIZA++ is created. As the final result of alignment, the obtained file must be automatically converted into the standard dictionary format by using my own created function called “GIZA++ to Dictionary format conversion”.

According to Table 1, the comparison between the aligned dictionary available in PASCAL website [9] and one obtained after following the process as shown in Figure 3, it discovers that our technique provides the aligned dictionary with higher quality because (1) the length of each word – letter sequence or phoneme string- is mostly the shortest after looking at Table 1 and (2) the number of corresponding phonemes of each alphabet -input letter-- is usually the smallest as shown in Figure 4.

For example, a letter ‘A’ in the original aligned dictionary can corresponds to 17 different phonemes as output such as /j/, /aJ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/, /æ/. Otherwise, 4 phonemes including /k/, /s/, /θ/, /θ/ are disappearing from the case of letter ‘A’ in our new aligned dictionary.

4. L2P conversion by using two-Stage NN

In this section, the most important techniques will be presented such as the LCD model, a combined technique between LCD and PCD models, a Phoneme Sequences Pattern-Observation model and the Two-Stage Neural Network based approach.

4.1. Letter Context-Dependent (LCD) model

In English, it seems to be very difficult to decide which output phoneme corresponds to the given letter because each letter can be mapped to various output phonemes in different contexts. By depending on the LCD model, each output phoneme can be predicted by observing on the sequence of connected letters (i.e. left context information + a mapping letter + right context information) rather than a single letter.

However, some problems are still happening after we measured the inconsistency of dictionary by respecting the LCD model [6]. It is still insufficient to solve such problems as shown in Figure 1, when the mapping letter ‘I’ on sequence “ACTICAL” can result more than one output phoneme /IH/ and /AH/ at the same time.
4.2. A combined technique between LCD and PCD models

Theoretically, a single letter can sometimes correspond to more than one output phoneme (’X’→K S) and vice versa. In order to cover all the mapping possibilities between letters and phonemes, the combined technique between LCD and PCD models in this paper allows (1) implementing the many-to-
many mapping concept between letters and phonemes in the system and (2) also increasing the number of distinct (letters, phonemes) pairs. This is the main key points to improve the performance of L2P conversion module [5].

|-------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|

4.3. Phoneme Sequences Pattern-Observation model

By looking carefully at the PCD column of both tables in Figure 6, the desired phoneme information, which is theoretically located at the central position of each corresponding phoneme sequence, is always found on the neighbor phoneme sequences as well (in a diagonal direction). Thus, each combination between current phoneme sequence (in Figure 8, known as Seq.) and its neighbor phoneme sequences (in Figure 8, known as Seq.-2, Seq.-1, Seq.+1 and Seq.+2) can generate a new pattern that can be recognized by the system. It is called Phoneme Sequences Pattern-Observation model which is very effective in solving the problem of conflicting phonemes as shown in Figure 1.

According to Figure 6, the phonemes that correspond to letter ’I’ on the three highlighted phoneme sequences of both words are unpredictable (replaced by *). It is because the same input letter sequence (i.e. ’ACTICAL’) can produce two different output phoneme sequences (i.e. /K T IH K AE/ and /K T AH K AE/) at the time, for example. On the contrary, this letter sequence can be generated into two different patterns after respecting the Phoneme Sequences Pattern-Observation model as presented in Figure 7.

4.4. Two-Stage Neural Network approach

The implementation of this new technique is based on the Two-Stage NN approach as described in Figure 8.

Following from left to right direction, a whole process of L2P conversion model is subdivided into two principal steps. The first-stage NN is for mapping from the sequence of M letters as input to N phonemes as output (More details is already described in section 4.2). Then, the second-stage NN is used to recognize the pattern obtained after applying the Phoneme Sequences Pattern-Observation model (section 4.3).
5. Experimental result

5.1. Datasets
Instead of using directly the original aligned CMUDict dataset [9] as other researchers did, we evaluated our new approach based on the new aligned CMUDict dataset. In this case, we used different alignment algorithm as described in Section 3. To ensure the high consistency of our CMUDict dictionary, we also deleted about 200 words including 28 words of a single letter, 157 acronyms, 8 numerical words and 7 special words (i.e. HABSBURG → JH AE P S B ER G). Moreover, there were about 300 words corrected manually by using only the regular expression searching function provided by Notepad++ editor. This new dictionary is divided into two parts; a training part contains 100,713 words (~90%) while a testing part contains only 11,188 words (~10%).

5.2. Configuration of FANN parameters
Each stage neural network has been created using the functions available in FANN (Fast Artificial Neural Network) library. After conducting many experiments, we have found that FANN outputs different results from one test to another depending on the configuration of its parameters. In our case, the best result has been given due to the following configurations:
- Standard FANN with 3 layers
- Number of neurons at hidden layer must be two times bigger or equal to that at input layer
- Learning rate = 0.8 ; Momentums = 0.1
- Training algorithm = INCREMENTAL
- Set_activation steepness_hidden = 1
- Set_activation steepness_output= 0.01
- fann_set_activation_function_hidden = FANN_SIGMOID_SYMOMETRIC_STEPWISE
- fann_set_activation_function_output = FANN_SIGMOID_STEPWISE
- Each letter is represented by 27 bits (including ‘_’) while each phoneme is represented by 40 bits.

5.3. Performance of our L2P conversion model
Focusing only on the improvement of Word Accuracy (WAcc) of the testing dataset (or OOV words), Table 2 shows that our two-stage NN approach provides at least 1.5% higher result in term of WAcc than the baseline approach (which is based only on a single neural network). Another point of view, the highest result is usually obtained once increasing the length of input letters at the first-stage NN and working on the high consistency dictionary.

If we are interested in the result based only on the seen words, our two-stage NN based approach is very powerful for modeling the L2P conversion module.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Input Letters</th>
<th>Length =7</th>
<th>Length = 9</th>
<th>Length =13</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OOV</td>
<td>Seen</td>
<td>OOV</td>
<td>Seen</td>
</tr>
<tr>
<td>Baseline 1 Phoneme</td>
<td>P Acc (%)</td>
<td>92.99</td>
<td>95.01</td>
<td>93.43</td>
</tr>
<tr>
<td>W Acc (%)</td>
<td>63.64</td>
<td>71.65</td>
<td>65.61</td>
<td>79.20</td>
</tr>
<tr>
<td>Two-Stage NN 5 Phonemes</td>
<td>P Acc (%)</td>
<td>93.33</td>
<td>97.55</td>
<td>93.91</td>
</tr>
<tr>
<td>W Acc (%)</td>
<td>65.20</td>
<td>85.33</td>
<td>68.51</td>
<td>87.65</td>
</tr>
</tbody>
</table>

Table 2: P Acc and W Acc of the OOV words and Seen words

6. Discussion
In order to improve the performance of L2P conversion model, the high consistency dictionary is always required. To create the aligned dictionary with high consistency, the many-to-many alignment concept and the idea that envisions a L2P alignment problem as a statistical machine translation problem are always recommended.

Otherwise, GIZA++ toolkit still needs some improvement in order to work perfectly with L2P alignment problem because there are still some errors occurred like “TAIPEI”→ NULL((5 6)) T((11)) AY((3)) P((4)) EY((2)), for example.

As a good part of our new approach, by observing very carefully on all the output results from both neural networks, the erroneous output phonemes given after the first-stage NN are also important because it can help in generating more different patterns before inputting into the second-stage NN.

7. Conclusions
In conclusion, this paper proposes a good idea that implements the many-to-many mapping concept into the L2P conversion model. Based on two-stage NN and focusing on both letter and phoneme context information, this approach is so simple and powerful since it can provide higher performance in term of phoneme as well as word accuracy if we compare with other previous approaches. Especially, it can solve effectively the problem of conflicting outputs at the phonemic level as described on the first page.

For future work, we will try to eliminate the mentioned problems by decomposing the dataset into two groups; one group contains only the words in which each vowel is surrounded only by the consonants (C1VC2) while another group contains all the remaining words (C1V1V2…C2).

8. References