A Multi-stage Method for Text-To-Pronunciation Conversion

Ching-Hsien Lee¹, Ren-Jr Wang¹, Chung-Jen Chiu¹

¹ Industrial Technology Research Institute, ICL, Bldg.51, 195, Sec.4, Chung Hsing Rd. Chutung, Hsinchu, 310
{C.H.Lee, RJWang, crchiu} @itri.org.tw

Abstract. Text-to-Pronunciation conversion is often used for speech synthesis and speech recognition-related systems. In this paper we present a data-driven, language-independent and multi-stage model for Text-to-Pronunciation conversion. With a Grapheme/Phoneme pair well aligned dictionary for training and utilizing a re-scoring strategy for those graphemes likely to be tagged erroneously, our model can not only increase the efficiency but also achieve a high accuracy than other data-driven approaches that have been applied to the same tasks.

Keywords: Text-To-Pronunciation, Grapheme, Phoneme.

1 Introduction

Text-to-Pronunciation (T2P) conversion convert input text (word) into output pronunciation, and is often used for speech synthesis and speech recognition-related systems. In fact, the best way to obtain the pronunciation of the input text is by looking into a dictionary. However, no dictionary will cover all the existing words and related pronunciations since new words have always been created as time goes on. Therefore the speech system may need a Text-to-Pronunciation conversion technique to generate the pronunciation for the text that is not collected within the dictionary. Since speech is a very important medium for human-computer interface, a good Text-to-Speech model is needed. Furthermore for some application domain like real-time speech synthesis, efficiency is also an important factor need to be considered. A T2P model with good accuracy and high efficiency is desired.

2 Related work

The traditional T2P conversion approach is rule-based, which encodes linguistic knowledge i.e. phonotactic, morphological knowledge as rules to convert words into their phonemic representations with reasonable accuracy [1][8][9]. However, such rules constructed by linguistic experts and can hardly been extended or maintained. Furthermore, as these rules differ for different languages, a huge amount of time and
human resources is required to construct new rules suitable for target language. Therefore, rule-based Text-to-Pronunciation conversion techniques lack reusability and portability.

Due to the shortcomings of rule-based approach, more and more Text-to-Pronunciation conversion model use data-driven methods, which include Chunk based method [2], Pronunciation by Analogy (PbA) [3], decision trees [4], Joint N-gram models [5][6], and HMM based [7]. The advantage of a data-driven method is that it does not require huge human resources and professional acknowledge and can be applied to different languages. Therefore, data-driven methods are better than rule-based methods.

Among the data-driven methods, PbA and Joint N-gram model are two of the most popular methods. The PbA method separates the input text into graphemes sequences with different lengths, and then compares these grapheme sequences with the text stored in the dictionary to find the possible phoneme sequences, and construct the grapheme and phoneme as a graph. The best path in the graph represents the pronunciation for the input text. The Joint N-gram model separates the text and pronunciation of an existing dictionary into grapheme-phoneme pairs, and the Joint N-gram model are estimated from the aligned dictionary using a standard maximum likelihood approach. Every input text is separated into possible grapheme-phoneme pairs and the grapheme-phoneme graph is established to find the best path through the Joint N-gram probability model.

Currently the Joint N-gram model has a higher accuracy, but efficiency decrease sharply while N>3. The PbA method is much more efficiently than the Joint N-gram model, but has a lower accuracy. Therefore in this paper we present a multi-stage Text-to-Pronunciation approach utilizing a re-scoring strategy to deal with those graphemes likely to be tagged erroneously. We can extend the graphemes that needed to be rescoring to pursuit higher accuracy or decrease some graphemes with low frequency to get better efficiency. So utilizing the re-scoring strategy we can find a balance between accuracy and efficiency.

3 A Multi-stage Text-to-Pronunciation model

There are three stages been applied in our Text-to-Pronunciation model. The first stage will segment the input text into grapheme sequence. Then each grapheme will be tagged with possible phonemes to construct a grapheme-phoneme pair graph. The top-N grapheme-phoneme pair sequence will be found as the output of second stage. The final stage will locate those graphemes likely to be tagged erroneously and extend the context range of those graphemes to extract more features (grapheme, phoneme or even grapheme-phoneme pair) to verify the possible phonemes at that location. The grapheme-phoneme sequence with highest score will be the conversion result. Fig1 is an overall picture of our conversion flow. In the following section we present how to get a well-aligned pronunciation dictionary, the theoretical formulation in each stage of our model, experiment flow and evaluation result comparing with existing model.
3.1 Dictionary preprocess

Before building a Text-to-Pronunciation model, the first thing is to obtain a training dictionary that every text and corresponding pronunciation should be well aligned. Here we use an iterative approach utilizing some well-defined grapheme-phoneme pairs in the documentation of CMU dictionary as seed to align the CMU pronunciation dictionary (http://www.speech.cs.cmu.edu/cgi-bin/cmudict). The alignment process is full automatically and can be easily applied to other language.

3.2 Stage 1: Grapheme Segmentation

In stage 1, an N-gram module is used to segment the input text into grapheme sequence $G=(g_1g_2...g_i...g_n)$ (where $g_i$ represent grapheme) according to the statistical information from our grapheme-phoneme aligned dictionary. The output will be at least one grapheme sequence. For example, if the input text is “feasible”, the possible grapheme sequence can be “f-e-a-s-i-b-l-e” or “f-ea-s-i-b-l-e”. For every grapheme sequence, the score $S_G$ can be obtained by the following formulation: where $n$ is the number of graphemes in the output grapheme-sequence.

$$S_G = \sum_{i=1}^{n} \log(P(g_i | g_{i-N+1}))$$  \hspace{1cm} (1)

3.3 Stage 2: Phoneme tagging

Here we choose the best grapheme-sequence from stage 1 output as the input in stage 2. According to the grapheme-phoneme map gather from the aligned dictionary, we can tag the possible phonemes for each grapheme and then establish a grapheme-phoneme pair graph. In order to find the best phoneme sequence, a score $S_P$ of every
phoneme sequence can be obtained by the following formulation: where $L$, $R$ represents ranges of a previous and following context of $g_i$, $n$ is the number of phonemes included in the phoneme sequence.

$$S_P = \sum_{i=1}^{n} \log(P(f_i | g_{i-L}^{i+R}))$$  \hspace{1cm} (2)

We extract at least Top-N phoneme sequence as the possible Text-to-Pronunciation conversion results, and their final scores $S_{G2P}$ are weight adjusted by the following formulation according to the stage1 and stage2 scores: where $S_G$ is the score of grapheme sequence, $S_P$ is the score of the phoneme sequence, and $W_G$ and $W_P$ are weight values for the $S_G$ score and $S_P$ score.

$$S_{G2P} = W_G S_G + W_P S_P$$  \hspace{1cm} (3)

Taking the grapheme-phoneme sequence with the highest score as a conversion result, when $L=1$, $R=2$, the word accuracy reach 59.71%, which already exceeds the average result of 58.54% for the PbA method. After detail analysis on the conversion result of stage2, we found that some of the graphemes are easily to be tagged erroneously, especially in vowels (such as a, e, i, o, u). The grapheme-phoneme map shows that the average number of phonemes corresponding to each vowel is 10.6. So the context range we choose in stage2 do not provide sufficient information for phoneme determination and may affect the accuracy. If we extend the context range in stage2 (for example, $L=3$, $R=3$), the search space will be enlarged and may decrease the efficiency.

In order to find a balance between accuracy and efficiency, we provide a re-scoring process that extend the context range search space only for those graphemes that are easily to be tagged erroneously.

### 3.4 Stage 3: Re-scoring

N grapheme-phoneme sequences with higher scores generated by stage2 are the input of this stage. Those graphemes that can cause phoneme determination error are located (currently, we are focusing on vowels only). The features sets $X(i)$ (may include graphemes($g$) as well as phonemes and grapheme-phoneme($\tau$) pairs, show as formulation 4) and the context range on every grapheme likely to be tagged erroneously are extended.
After extracting new features for every grapheme likely to be tagged erroneously, we use mutual information (MI) to calculate the connection between those features and the phonemes corresponding to the grapheme likely to be tagged erroneously. The mutual information indicates a possibility of the features and corresponding phoneme showing together, and the re-scoring score is obtained to the grapheme-phoneme sequence as follows: where \( w_j \) is a weight value, \( E \) is a set of graphemes likely to be tagged erroneously (the set can be extend due to the practical issue).

\[
X(i) = \bigcup_{n=1}^{N} X_n(i, g) \bigcup_{n=1}^{N} X_n(i, f) \bigcup_{n=1}^{N} X_n(i, r)
\]

\[
X_n(i, y) = \{ x | x = y_j \ldots y_r, i - L \leq l \leq i + R \land (r - l + 1) = n \land i \notin [l, r] \}
\]

After re-scoring n best grapheme-phoneme sequences from stage2, a re-scored score \( S_R \) of every grapheme-phoneme sequence is obtained. Finally the weight adjusted \( S_{G2P} \) score are combined to obtain a final score:

\[
S_{Final} = w_{G2P} S_{G2P} + w_{R} S_{R}
\]

\[ (5) \]

4 Experiment

In order to prove the great improvement of Multi-stage Text-to-Pronunciation model both in performance and efficiency, we use CMU pronunciation dictionary as our experimental data. The CMU pronunciation dictionary is a machine-readable dictionary, which includes over 125,000 English words and corresponding pronunciations. These pronunciations are composed of a phoneme set including 39 phonemes. After taking out pronunciation marks and words with multiple pronunciations, 110,327 words remains. All graphemes \( G(w) = g_1g_2...g_n \) of each word \( w \) and corresponding phonemes \( P(w) = f_1f_2...f_m \) are aligned automatically through our iterative alignment process and grapheme-phoneme sequences \( GP(w) = g_1p_1: g_2p_2: ...: g_mp_m \) is obtained. Afterward, all grapheme-phoneme sequences are randomly divided into ten sets, and then cross-validation is performed for an experimental evaluation. We take both Joint N-gram and PbA model as our baseline approach.
In our multi-stage approach, the experiment shows that the word accuracy reaches 59.71% after first two stages been applied. After extending context range from L=1, R=2 to L=5, R=5 in the re-scoring stage, the overall accuracy can achieve 69.13% (with an error reduction rate of 23.38%). The accuracy exceeds Joint N-gram model accuracy of 68.3% (N=4) and the efficiency is also faster than the Joint N-gram model (N=3,4). Overall comparison shows in Fig2, 3.

<table>
<thead>
<tr>
<th>Model</th>
<th>Word Accuracy</th>
<th>Phoneme Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PbA</td>
<td>58.5%</td>
<td>89.0%</td>
</tr>
<tr>
<td>Joint 3-gram</td>
<td>64.5%</td>
<td>91.2%</td>
</tr>
<tr>
<td>Joint 4-gram</td>
<td>68.3%</td>
<td>92.0%</td>
</tr>
<tr>
<td>Our approach</td>
<td>69.1%</td>
<td>92.3%</td>
</tr>
</tbody>
</table>

Fig. 2. The overall comparison with Joint N-gram and PbA

Fig. 3. The efficiency comparison between Multi-stage model and Joint N-gram model

For a further analysis as show in Fig4., the average accuracy of vowel phonemes is also raised from 69.72% to 81.16%, with an error reduction rate of 37.78%. Consequently, the multi-stage approach can improve the accuracy of Text-to-Pronunciation conversion.
5 Conclusion

We propose a multi-stage Text-to-Pronunciation conversion with a re-scoring stage for those graphemes likely to be tagged erroneously. An experiment result shows that our accuracy outperforms Joint N-gram and is 13.67 times the conversion speed of it (N=4). We believe this approach can find a balance between accuracy and efficiency comparing to existing data-driven approaches.

6 Acknowledgements

This paper is a partial result of Project 5352B23100 conducted by ITRI under sponsorship of the Ministry of Economic Affairs, R.O.C..

References


Fig. 4. The error rate comparison while using re-scoring stage

