Methods for Efficient Semi-Automatic Pronunciation Dictionary Bootstrapping

Tim Schlippe, Matthias Merz, Tanja Schultz

Cognitive Systems Lab, Karlsruhe Institute of Technology (KIT), Germany
tim.schlippe@kit.edu

Abstract

In this paper we propose efficient methods which contribute to a rapid and economic semi-automatic pronunciation dictionary development and evaluate them on English, German, Spanish, Vietnamese, Swahili, and Haitian Creole. First we determine optimal strategies for the word selection and the period for the grapheme-to-phoneme model retraining. In addition to the traditional concatenation of single phonemes most commonly associated with each grapheme, we show that web-derived pronunciations and cross-lingual grapheme-to-phoneme models can help to reduce the initial editing effort. Furthermore we show that our phoneme-level combination of the output of multiple grapheme-to-phoneme converters reduces the editing effort more than the best single converters. Totally, we report on average 15% relative editing effort reduction with our phoneme-level combination compared to conventional methods. An additional reduction of 6% relative was possible by including initial pronunciations from Wiktionary for English, German, and Spanish.

Index Terms: semi-automatic pronunciation generation, pronunciation modeling, web-derived pronunciations, phoneme-level combination

1. Introduction

From some 7,100 languages all over the world, only a small fraction of economically relevant languages are covered by data resources needed for speech technologies like Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) systems. These resources include text corpora, transcribed speech data and pronunciation dictionaries. The latter provide a mapping from the written form of a word to its pronunciation, typically expressed as a sequence of phonemes. Automatic methods for grapheme-to-phoneme (G2P) conversion being able to infer pronunciations are of great help in the process of generating dictionaries. Since these methods need knowledge about the target language either in the form of pronunciation rules or as sample dictionary entries, they are not applicable to bootstrap dictionaries for languages where such data is not available or too expensive to generate. However, they help in semi-automatic strategies where the generated pronunciation hypotheses are manually checked and corrected by humans [1, 2, 3, 4].

In this paper we present the following contributions to the rapid and economic semi-automatic dictionary development: For an efficient semi-automatic pronunciation generation process, the G2P model has to be updated regularly. We evaluate different intervals for these updates and propose an optimization. Common approaches utilize one G2P conversion tool of their choice. We combine the output of different converters to improve the accuracy of the created hypotheses and thus further lower the editing effort. Usually the bootstrapping process is started with an empty dictionary or a set of manually created word-phoneme (W-P) pairs. In our strategy we integrate additional pronunciations to reduce the initial editing effort: Pronunciations created by the concatenation of single phonemes most commonly associated with each grapheme (1:1 G2P mapping), web-derived pronunciations (WDP) and pronunciations derived from G2P models of other languages (cross-lingual).

The Rapid Language Adaptation Toolkit (RLAT) [5] is a freely available online service to reduce the amount of time and effort involved in building speech processing systems for new domains and languages. We included our pronunciation generation strategy into RLAT.

2. Related work

In semi-automatic dictionary generation processes like [1], [2], and [3] native speakers enter pronunciations as phoneme strings. To reduce the difficulty of pronunciation generation, the user can listen to a synthesized wavefile of the entered pronunciation. Like [3], we present a list of available phonemes to the users, automatically reject pronunciations containing invalid phoneme labels and enable the user to listen to a synthesized wavefile of the pronunciation.

[1] and [2] display the words according to their occurrence frequencies in a text corpus. By covering common words early, word error rates (WERs) in ASR are minimized for an early ASR training and decoding. [4] and [6] order the words according to their n-gram coverage to learn many G2P relations early. [7] and [8] prefer short words over longer words to alleviate the correction effort for human editors. We follow the principles of [4] and [6] and additionally prefer short words over longer words like [7] and [8]. While [2] use a phoneme set defined by linguists, [4] infers a phoneme set in an automatic way: An initial phoneme recognizer is trained on a grapheme-based dictionary. Based on audio recordings and transcriptions, acoustic model units are adapted based on Merge&Split. In [3] and in our approach no additional audio recordings are required since users manually specify the phonemes from an International Phoneme Alphabet (IPA) chart, guided by language-independent audio recordings of each phone.

Some approaches update the G2P model after each word [2, 3]. Others combine incremental updates and periodic rebuilding [4, 9]. In [1] and [10], the user decides about the creation of new G2P models. [9] introduce a data-adaptive strategy, updating the G2P model after the pronunciations of 50 words needed to be corrected. While [2] start with an empty dictionary, [1] manually generate pronunciations for the most frequent 200-500 words in a text corpus. [3] initializes the pronunciation generation with a 1:1 G2P mapping entered by the

\footnote{http://csl.ira.uka.de/rlat-dev}
user. [4] records 20 minutes of speech and builds an initial dictionary automatically based on the grapheme-based phoneme set, acoustic information and their transcriptions. Since web-derived pronunciations proved to be helpful for the dictionary generation process [11, 12], we used them to obtain initial training data. While conventional approaches use only one G2P converter, we use multiple G2P converters with similar performances and combine their outputs [13].

3. Semi-automatic pronunciation generation strategy

Fig. 1 illustrates our strategy. The components where our methods deviate from state-of-the-art are highlighted. Starting with an empty dictionary, our process consists of the following steps:

1. Initial W-P pairs are used to train an initial G2P model.
2. The next word is determined.
3. Each G2P converter generates a hypothesis for the pronunciation of that word.
4. The hypotheses are combined at a phoneme level combination, which produces one hypothesis to be presented to the user.
5. Optionally, the 1st-best hypotheses of each G2P converter are additionally offered to the user.
6. The user edits the best-matching hypothesis to the correct pronunciation for the requested word.
7. Word and corrected pronunciation are added to the dictionary.
8. After a certain number of words, the G2P converters update their G2P models based on the W-P pairs in the dictionary.

4. Experimental setup

4.1. Languages and reference dictionaries

Since our methods should work for languages with different grade of regularity in G2P relationships, our experiments are conducted with German (de), English (en), Spanish (es), Vietnamese [14] (vi), Swahili (sw), and Haitian Creole (ht). For evaluating our G2P conversion methods, we use GlobalPhone dictionaries [15] for de, es and sw as reference since they have been successfully used in LVCSR [16]. The en dictionary is based on the CMU dictionary. For ht, we employ a dictionary also developed at CMU. All dictionaries contain words from the broadcast news domain. We created six random excerpts of 10k words from each dictionary to conduct all experiments in a 6-fold cross-validation, except for vi as the Vietnamese dictionary contains only 6.5k word-phoneme pairs.

4.2. G2P converters

We use four G2P converters for our experiments: Sequitur G2P [17], Phonetisaurus [18, 19], Default&Refine [20, 21, 22] and CART trees [23]. For all G2P converters, we use context and tuning parameters that result for all six tested languages in an optimal tradeoff between performance and CPU time for the G2P model training.

4.3. Evaluation metrics

As it is very expensive to assess real human editing times, we simulate the annotation process assuming that the editor changes the displayed phoneme sequence to the phoneme sequence of the reference pronunciation. To measure the human editing effort, we introduce the cumulated phoneme error rate (cPER) as follows:

\[
cPER(n) := \frac{\sum_{i=1}^{n} (\text{sub}(w_i) + \text{ins}(w_i) + \text{del}(w_i))}{\sum_{i=1}^{n} \text{phonemes}(w_i)}
\]

We accumulate the number of edits (substitution, insertion or deletion of single phonemes) a developer would have done up to the current word \(w_n\) to reach the pronunciations of our reference dictionaries and set these edits in relation to the total number of phonemes seen in the dictionary. This value is the counterpart to the phoneme correctness in [22]. As the values contain the initialization phase with bad hypotheses, reading these numbers as PER which reflects only the editing effort for the current word \(w_n\) would be misleading. According to [7], we assume a human dictionary developer to take 3.9 seconds on average for an edit to a predicted hypothesis.

5. Experiments and results

5.1. Word selection strategy

Based on [4] and [6], our word list is sorted by graphemic 3-gram coverage and based on [8] with a preference for short words (ngram sorted) to speed up the annotation process.

Figure 2: Word selection strategies, evaluated on 10k English W-P pairs with Phonetisaurus.

Fig. 2 shows that our proposed strategy outperforms a random order slightly in cPER for English dictionary extracts. Like [6], we plot an alphabetical selection for comparison. The impact of ngram sorted is higher in the beginning of the process, when less training data for the G2P models are given. In all three curves we updated the G2P model according to logistic growing intervals which we describe in Sec. 5.2. ngram sorted outperforms random and alphabetical for the other languages as well.
5.2. Iterative G2P model building

The more frequent G2P models are re-created based on the incremental pool of W-P pairs, the better the quality of the generated pronunciations which reduces the human editing effort. However, frequent G2P model generation results in a notable increase in CPU time. For example, the slowest G2P converters in our selection take approximately one hour for a G2P model re-training pass of 10k en W-P pairs on a computer equipped with a 2.6GHz AMD Opteron processor. Since parallel or incremental G2P model training is for some of the methods not possible, our goal is to train G2P models more frequently in the beginning when it does not take much time and G2P model quality still increases rapidly with more training data. [9] proposes a data-adaptive training interval (Adaptive). In a first phase they re-train their G2P model after each added word. When the dictionary reaches a size of 1,500 words, they switch to the second phase where the G2P model is re-created after 50 predicted pronunciations needed corrections. We compared Adaptive to a re-training at a fixed dictionary growth (Fixed) and linearly growing intervals (Linear) with different parameter values. 10% dictionary growth proved to be a sensible value for Linear with better results in less time than Fixed. However, Linear exhibits the disadvantage of a boundless increase of the training intervals for large dictionaries. To ensure a maximum size for the interval, we propose a logistic growth function (Logistic). This function starts with training interval 1 after word 1 and enables a smooth increase from 0 to 10k words where we observed a notable slowdown in G2P model improvement even for the languages with a high complexity in the G2P relation. In our case we limit the maximum training interval to 3k words.

Table 1: Strategies for G2P model retraining.

<table>
<thead>
<tr>
<th>Interval</th>
<th>sum</th>
<th>edit time</th>
<th>CPU time</th>
<th>∑ Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic</td>
<td>22,983</td>
<td>89,634 s</td>
<td>2,055 s</td>
<td>91,689 s</td>
</tr>
<tr>
<td>Linear</td>
<td>22,657</td>
<td>88,362 s</td>
<td>3,282 s</td>
<td>91,644 s</td>
</tr>
<tr>
<td>Adaptive</td>
<td>22,530</td>
<td>87,867 s</td>
<td>28,928 s</td>
<td>116,795 s</td>
</tr>
<tr>
<td>Fixed</td>
<td>23,658</td>
<td>92,266 s</td>
<td>17,529 s</td>
<td>109,795 s</td>
</tr>
</tbody>
</table>

Table 2: cPERs (%) for single G2P converters, 10k dictionaries

5.4. Resources for initial pronunciations

In addition to the traditional substitution of graphemes with the most commonly associated phonemes (1:1 G2P Mapping), we show that G2P models from web data and from other languages can help to reduce the initial human editing effort.

5.4.1. 1:1 G2P mapping

As in [3], we created initial pronunciations with 1:1 G2P Mapping. This mapping can be compiled by a native speaker but also derived from existing W-P pairs, e.g. from the Web. How close the pronunciations with the 1:1 G2P Mapping come to our validated reference pronunciations in terms of PER is illustrated in Tab. 3. Including the pronunciations generated with the 1:1 G2P Mapping in the PLC with the single G2P converter outputs helps to reduce the cPER for the first 100 en words, the first 170 de words, the first 360 es words, the first 80 vi words, the first 230 sw words, and the first 120 lt words (Optim. x-over). Using the pronunciations from the 1:1 G2P Mapping after these crossovers reduces the pronunciation quality in the PLC. Therefore in our strategy we use the pronunciations from the 1:1 G2P Mapping in the first place in the PLC up to the average crossover of all tested languages at 180 words and omit them afterwards. Despite the high PERs in the pronunciations from the 1:1 G2P Mapping, we obtain on average a relative cPER reduction of 3% on top of the PLC as shown in Tab. 4.

5.4.2. Web-driven G2P converters’ output

Since web-derived pronunciations (WDPs) proved to support the dictionary generation process [11, 12, 13, 25], we investigated if they can be used to obtain initial training data for our G2P converters and outperform the conventional 1:1 G2P Mapping. For our analysis we used Sequitur to build additional G2P models to which we compare all improvements. For the phoneme-level combination (PLC) [13], we apply nbest-lattice at the phoneme-level which is part of the SRI Language Modeling Toolkit [24]. From each G2P converter we select the most likely output phoneme sequence (1st-best hypothesis). Then we use nbest-lattice to construct a phoneme lattice from all converters’ 1st-best hypotheses and extract the path with the lowest expected PER. Since we observed that the order of pronunciations is of great importance for the results, we ordered the 1st-best hypotheses according to the average performance of the different G2P converters in our baseline scenario: Sequitur, Phonetisaurus, D&R, CART tree. As demonstrated in Tab. 2, PLC leads to a statistically significant reduction in cPERs (ΔPLC) for all languages between 1.9% and 38.1% relative.

Table 3: PER (%) and optimal cross-over for initial pron.

<table>
<thead>
<tr>
<th>PER 1:1</th>
<th>en</th>
<th>de</th>
<th>es</th>
<th>vi</th>
<th>sw</th>
<th>lt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optim. x-over</td>
<td>50.01</td>
<td>37.83</td>
<td>14.20</td>
<td>40.49</td>
<td>10.52</td>
<td>14.92</td>
</tr>
<tr>
<td>PER Wiki</td>
<td>52.55</td>
<td>33.47</td>
<td>11.40</td>
<td>21.40</td>
<td>23.10</td>
<td>23.10</td>
</tr>
<tr>
<td>Optim. x-over</td>
<td>49.02</td>
<td>38.46</td>
<td>13.02</td>
<td>21.40</td>
<td>23.10</td>
<td>23.10</td>
</tr>
<tr>
<td>PER x-lingual</td>
<td>42.42</td>
<td>46.64</td>
<td>13.02</td>
<td>21.40</td>
<td>23.10</td>
<td>23.10</td>
</tr>
<tr>
<td>Optim. x-over</td>
<td>42.42</td>
<td>46.64</td>
<td>13.02</td>
<td>21.40</td>
<td>23.10</td>
<td>23.10</td>
</tr>
</tbody>
</table>
converters for en, de and es with W-P pairs from Wiktionary\(^3\). How the WDPs approach our reference pronunciations in terms of PER is illustrated in Tab. 3. Including the WDPs in the PLC benefited for the first 210 en words, the first 4,750 de words, and the first 230 es words (Optim. x-over). Instead of omitting them, we gained more cPER reduction by putting the WDPs from the first to the last position in the PLC after the average crossover of all tested languages at 500 words. As demonstrated in Tab. 4, using the web-driven G2P converters’ output to reduce the initial effort performed better than the 1:1 G2P Mapping with a relative improvement in cPER of 6% compared to the PLC. Applying our automatic filtering methods which had further improved the quality of web-driven G2P converters in [26] and [27] did not lower the cPER. The reason is that the filtering skips irregular pronunciations from the input which have supplied valuable additional information to the PLC.

5.4.3. Cross-lingual pronunciations

In [28] we have shown that using G2P models derived from existing dictionaries of other languages can severely reduce the necessary manual effort in the dictionary production, even more than with the 1:1 G2P Mapping. According to the cross-lingual dictionary generation strategy in [28], we (1) mapped the target language graphemes to the graphemes of the related language, (2) applied a G2P model of the related language to the mapped target language words, and (3) mapped the resulting phonemes of the related language to the target language phonemes. With this strategy, we generated en pronunciations with a de G2P model and de pronunciations with an en G2P model. How close the cross-lingual pronunciations come to our reference pronunciations in terms of PER is illustrated in Tab. 3. Including the cross-lingual pronunciations in the PLC with the single G2P converter outputs helped slightly for the first 42 en words and the first 52 de words (Optim. x-over). Therefore in our strategy we use those pronunciations in the first place in the PLC up to the average crossover of all tested languages at 45 words and omit them afterwards. While we observe a small relative cPER reduction of 0.34% on top of the PLC for en, we obtain a relative increase of 0.56% for de as shown in Tab. 4. However, Fig. 3 demonstrates that cross-lingual outperforms 1:1 G2P Mapping in the beginning of the process, when less training data for the G2P models are available.

6. Conclusion and future Work

We have proposed efficient methods for rapid and economic semi-automatic dictionary development and evaluated them on languages with different G2P relation complexity. We measured the human editing effort with the cPER, setting the number of edits done by a developer up to the current word in relation to the total number of reference phonemes. Tab. 4 summarizes the cPERs and the necessary edits for 10k en, de, es, sw, and ht and 6.5k vi words. While for the languages with a strong G2P relationship only a few hundred edits are required for all words, and for Spanish between 1.5k and 1.8k, we observe almost 10k required edits for de and en. In Fig. 3 we have plotted the cPER reduction over the number of processed pronunciations for en, the language with the highest G2P complexity. Our word selection strategy ngram sorted outperforms random. Updating the G2P model according to logistic growing intervals enables between 7% and 60% CPU time savings with performances comparable to other approaches. Our PLC of the output of multiple G2P converters reduces the editing effort by on average 15% relative to the best single converter, even 38% for sw. The traditional 1:1 G2P Mapping helps de and en with complexer G2P relationships only slightly to reduce the editing effort. cross-lingual only outperforms 1:1 G2P Mapping in the beginning of the process, when less training data for the G2P models are available. However, we recommend to use WDPs on top of PLC if available, since they give us consistent improvements for different vocabulary sizes in the whole process and on average 6% relative for 10k words. Our new RLAT function which is publicly available allows to bootstrap a dictionary with the proposed methods supported with the possibility to listen to a synthesized waveform of the pronunciation. Future work may include to analyze our strategy in a crowdsourcing scenario and for other languages. Furthermore, our word selection strategy may be further improved with active learning techniques [29].

---

\(^3\)http://www.wiktionary.org
7. References


