On the Importance of Exception and Cross-word Rules for the Data-driven Creation of Lexica for ASR

Qian Yang and Jean-Pierre Martens

ELIS-RUG, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
Phone: +32 (0)9 264 3397  Fax: +32 (0)9 264 3594
E-mail: {qianyang,martens}@elis.rug.ac.be

Abstract—Based on earlier work [1], we developed [2] a new data-driven approach for building a lexicon with multiple pronunciation variants per word. The method automatically learns stochastic pronunciation rules that are then used to transform a reference pronunciation (e.g. taken from a pronunciation dictionary) into a list of pronunciation variants. The results obtained with the new approach were quite spectacular: the word error rate on TIMIT could be reduced by more than 45% in a closed vocabulary situation.

During the development of our system we argued for the need of cross-word rules and exception rules. The latter prohibit rather than generate a pronunciation variant in a particular situation. In this contribution we describe experiments that assess the importance of these two rule types. The results indicate that by ignoring the cross-word rules, about 40% of the benefit of our approach is lost. On the other hand, some reduction of the exception rule importance seems to be quite acceptable.

We can now propose a new rule learning system that has become 9 times faster and is yet offering about the same performance.

Keywords—lexical modeling; cross-word rules; exception rules; ASR system

I. INTRODUCTION

It is acknowledged that automatic speech recognition (ASR) would benefit from an improved lexical component [5]. This component describes the likely phonetic transcription of the words to be recognized. The standard lexicon normally consists of a single pronunciation per word. However, the existence of pronunciation variants, e.g. due to cross-word coarticulations, is well known. Therefore, one may expect to obtain a better ASR performance by using a lexicon with multiple pronunciation variants per word.

Recently we proposed [2] a new data-driven approach for building such a lexicon. It consists of two main steps:

1. A set of stochastic pronunciation rules are learned automatically from an orthographically transcribed corpus, by means of a four-phase process.
2. Each reference word transcription from the standard lexicon is transformed into a list of word pronunciation variants by means of these rules.

With this new approach, large relative reductions of the word error rate on TIMIT were measured: as large as 45% for the closed vocabulary and 20% for the open vocabulary situation.

II. LEARNING PRONUNCIATION RULES

The pronunciation rules in our approach are described in the following format:

\[ r : LFR \rightarrow F' \]  

with \( L, F, R \) and \( F' \) representing variable length phoneme strings. The meaning of the rule is that a focus \( F \) surrounded by left and right contexts \( L \) and \( R \), can be transformed to an output \( F' \) with a firing probability \( P_{fr} \). In what follows \( (F, F') \) and \( LF \) will often be called the transformation and the condition of the rule. Important is that the rules are collected in the rule set that is governed by a rule selection hierarchy [2].

The rule learning process consists of the following four phases:

1. Generate reference and expert transcriptions of the training data, insert word boundaries between consecutive word transcriptions, and line-up the expert with the reference transcriptions.
2. Identify candidate rules in two steps:
   (a) Collect all transformations \( (F, F') \) in the alignment of the expert and reference transcriptions.
   (b) For each occurrence of a transformation, collect different left and right contexts (= different number of considered phonemes) to construct a set of rule conditions.
3. Learn the firing probabilities of the rules. Two counters \( n_1(r) \) and \( n_2(r) \) are associated with each rule: \( n_1(r) \) is the number of times the rule condition was selected (taking the rule hierarchy into account), and \( n_2(r) \) is the number of times the rule was used to transform the reference transcription. The firing probability is then obtained as \( P_{fr}(r) = n_2(r)/n_1(r) \).
4. Prune the rule set according to some statistical criterion, until the amount of captured knowledge starts to decrease significantly.

The following basic principles were adopted in this learning process:

1. Any transformation of subsequent phonemes of the reference transcription should normally be considered as being performed by a single rule.
2. Any focus consisting of more than \( N_F \) phonemes is considered invalid.
3. Any transformation that would induce the deletion of an entire word is considered invalid as well.
4. As it will not be possible to explain all observed pronunciation variations by means of a concise rule set,
transformations occurring too infrequently (less than $N_{\text{trans}}$ times) are removed after the first step in phase 2.

5. The condition of a rule is not allowed to extend over more than two consecutive words.

6. The left/right context of a rule is not allowed to comprise more than $N_{LR}$ phonemes (word boundaries excluded).

III. Pronunciation Variant Generation

During the variant generation process, the rule firing probabilities give rise to variant probabilities. By eliminating all variants with probability less than $P_{\text{min}}$, one can easily control the number of variants being produced on the basis of a rule set.

During the development of our system we argued for the importance of cross-word and exception rules. A cross-word rule is one whose condition extends over more than one word. Exception rules (also called negative rules) are rules with a low firing probability. They tend to prohibit the generation of a variant in a particular situation. Another issue we want to address is the importance of the variant probabilities during recognition.

The main goal of the work described in this paper is therefore to discover what performance gains or losses one can expect from prohibiting cross-word and exception rules, and from ignoring the variant probabilities during recognition. The rest of the paper is organized as follows: section IV reviews the experimental procedure we adopted, section V describes the prohibition of cross-word rules, section VI that of exception rules, and section VII the ignorance of the variant probabilities during recognition.

IV. Validation Experiments

A. Training and testing material

All experiments described in this paper were carried out on the TIMIT database [3]. It contains a total of 6300 sentences: 630 speakers from 8 major dialect regions in the United States each read 10 sentences. The text material in TIMIT consists of 2 dialect sentences (SA sentences), 450 phonetically-compact sentences (SX sentences) and 1890 phonetically-diverse sentences (SI sentences). Each speaker read the 2 SA sentences, 5 SX sentences and 3 SI sentences. As the SA sentences were only meant to expose the dialectal variations among speakers, we did not consider them during training nor testing. The SX sentences imply a vocabulary of 1793 words, the SI sentences one of 5143 words. The total vocabulary (SX+SI) counts 6224 words. A bigram that models all the TIMIT sentences was used as the language model [1].

The database was divided into a training and a test database (462 speakers for training and 168 for testing), and the test database was further divided into the core test set and the rest. All experiments described here were performed on the core test set which was further subdivided in an SX and an SI part.

B. ASR system configuration

The ASR engine was a segment-based recognizer incorporating context-independent discriminatively trained acoustic models [4]. The lexicon was either composed of one pronunciation per word or multiple pronunciations per word as they were created by means of pronunciation rules. In both cases, the word models were supplemented with deletion, insertion and substitution transitions. These so-called error arcs were characterized by error probabilities that were trained automatically.

The acoustic models were trained on all the SX+SI training sentences, the pronunciation rules on all the SX training sentences, and the error probabilities on half of these sentences. By doing so, it is possible to investigate whether rules learned on one vocabulary (1793 words) can be used to transform another vocabulary (5143 words). We thus discern the open vocabulary situation in which we train on SX and test on SI, and the closed vocabulary situation in which training and testing are both done on SX.

C. Baseline system

In the baseline system (BS), the single pronunciation per word was taken from the lexicon distributed together with the database. The recognition results using these reference pronunciations are listed in Table I.

<table>
<thead>
<tr>
<th></th>
<th>SX+SI</th>
<th>SX</th>
<th>SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>8.41</td>
<td>4.03</td>
<td>14.31</td>
</tr>
</tbody>
</table>

D. Full-scale rule learning systems

The so called full-scale rule learning systems consist of a lexicon that was constructed using rules that were learned by the full-scale algorithm described in [2]. By changing the settings of the thresholds, the size of the learned rule set can be manipulated. For one setting, we obtained a rule set of 1181 rules. The emerging WERs (relative to the baseline WER) are depicted as a function of the number of variants per word (controlled on the basis of $P_{\text{min}}$) on Figure 1.

For the closed vocabulary situation, the improvement over the baseline is maximally 47%. For the open vocabulary situation, it is of the order of 20%.
V. Prohibition of Cross-word Rules

A. Cross-word rules

It is generally acknowledged that words in continuous speech do not sound as words spoken in isolation. There are a number of phonological phenomena that are particularly relevant at the word boundaries. Two of them are degemination and connection sound insertion.

If the final phoneme of the first word and the initial phoneme of the next word are identical, they fuse to one phoneme when these two words are occurring in fluent speech (e.g. /now we go/ becomes /nowe go/). This phenomenon is called degemination.

In case a word ends on a vowel and the next word starts with another vowel, it often happens that an extra connection phoneme (usually a /w/ or /j/) is inserted between those words in fluent speech (e.g. /you are/ becomes /you-w-are/).

We argue that these phenomena must be taken into account during recognition. Since they can only be described naturally by means of cross-word rules, we expect that a rule learning system accommodating cross-word rules will perform better than one not accommodating them. On the other hand, prohibiting cross-word rules to occur in the rule set makes the recognition simpler and more efficient because the created pronunciation variants become context-independent then.

Taking the above into account, it is interesting to investigate whether the loss in performance due to the prohibition of cross-word rules balances the gain in simplicity/efficiency of the recognizer.

B. Prohibition of cross-word rules

In order to obtain the optimal performance of an ASR with context-independent word pronunciation models, we did not proceed by eliminating the cross-word rules from a learned rule set. On the contrary, we configured our rule learning system in such a way that the left and right contexts selected in phase 2 were not allowed to cause the rule condition to extend over more than one word. The rule condition was allowed though to be delimited by one or two word boundary symbols. In that case we refer to the corresponding rule as a word boundary rule.

C. Experimental results

We have compared pairs of ASR systems that were obtained using the same threshold settings in the rule learning process, but with a cross-word rule switch set ON and OFF respectively. In Table II, we have listed results for three such system pairs (different threshold settings). We used the same value of \( P_{min} \) in all cases.

By comparing these recognition results with the ones in Table I, it follows that for the closed vocabulary situation between 29% and 39% of the benefit of our approach got lost. For the open vocabulary situation, this loss was between 11% and 54%. The total performance gain (SX+SI) is reduced by 24...45%. The number of rules being generated is not very much affected by the extra constraint on the rule condition.

D. Conclusion

The data clearly indicate that cross-word rules are very important in our pronunciation modeling approach.

VI. Reducing the Importance of Exception Rules

A. Proposed Mechanism

After phase 2 of the rule learning process we have collected a large list of candidate rules \((\text{LFR}, F')\). For each rule we have also counted the number of times the rule condition was satisfied in the reference transcription, while the focus \( F \) was transformed to \( F' \) in the expert transcription.

The idea is now to eliminate a number of candidate rules on the basis of their counters. However, by doing so, we will prune rules whose condition does not often occur in conjunction with the transformation \((F, F')\) implied by the rule. Strictly speaking, this is not a sufficient condition for pruning a rule since it is still possible that the same rule condition would be observed many times in situations where the focus is left unaltered. In that case \((\text{LFR}, F')\) should be considered as an exception rule for the transformation \((F, F')\). On the other hand, removing a large number of rule candidates before entering the third phase of the rule learning process can save a lot of CPU-time. Therefore, it is worthwhile to investigate the effects of this additional pruning stage on the quality of the final rule set.

As we were reluctant at first to introduce this pruning stage, we designed it in such a way that rules with a low count \((< N_{cm})\) were only pruned if the number of context symbols (word boundaries included) in the rule condition was sufficiently large \((> N_{cont})\).

B. Experimental results

We used one setting of the standard thresholds of the rule learning process, namely the setting that yielded a rule set of 1181 rules when the full-scale rule learning was applied. Then we examined what happens if the proposed pruning is inserted. In particular, we investigated how the ASR performance depends on \((N_{cm}, N_{cont})\). Clearly, if \( N_{cm} \) is much larger than \( N_{trans} \) (the minimum number of times a transformation \((F, F')\) must be observed to be retained in phase 2), many pairs otherwise surviving in phase 2 will be pruned during this extra rule pruning stage.

If it is much smaller than \( N_{trans} \) then too few rules will be pruned. Therefore, we chose to set \( N_{cm} = N_{trans} \).

We performed experiments for \( N_{cont} = 0, 1 \) and 2. The larger \( N_{cont} \) is, the fewer candidate rules are pruned and the more rules are retained in the final rule set: 1021 rules when \( N_{cont} = 2 \), 715 when \( N_{cont} = 1 \) and 548 when \( N_{cont} = 0 \).

### Table II

<table>
<thead>
<tr>
<th>nr of rules</th>
<th>cross word</th>
<th>SX+SI</th>
<th>SX</th>
<th>SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>2289</td>
<td>Yes</td>
<td>6.28</td>
<td>2.83</td>
<td>10.89</td>
</tr>
<tr>
<td>2116</td>
<td>No</td>
<td>7.25</td>
<td>3.18</td>
<td>12.73</td>
</tr>
<tr>
<td>1181</td>
<td>Yes</td>
<td>6.15</td>
<td>2.12</td>
<td>11.59</td>
</tr>
<tr>
<td>1113</td>
<td>No</td>
<td>6.70</td>
<td>2.87</td>
<td>11.87</td>
</tr>
<tr>
<td>628</td>
<td>Yes</td>
<td>6.22</td>
<td>2.55</td>
<td>11.16</td>
</tr>
<tr>
<td>625</td>
<td>No</td>
<td>6.89</td>
<td>3.08</td>
<td>12.02</td>
</tr>
</tbody>
</table>
The CPU-time required for producing the final rule set is dramatically affected by the extra rule pruning stage. The speed-up factor ranges from 4 \((N_{cont} = 2)\) to 9 \((N_{cont} = 0)\).

The ASR performances obtained with lexica built on the basis of these three different rule sets are depicted on Figure 2. The WERs were measured as a function of the average number of variants per word for each rule set.

C. Conclusion

The data show that by reducing the importance of exception rules, the rule learning process can be speeded up dramatically (9 times faster), the number of rules to consider during the variant generation process can be reduced significantly (factor 2), and the ASR performance can be maintained. Although some of the performance benefit was lost in the closed vocabulary case, it was recovered in the open vocabulary situation. This means that the new system generalizes better to other vocabularies.

VII. Ignoring the variant probability during recognition

The variant generation process produces pronunciation variants with attached probabilities which sum up to 1. One may wonder how important these probabilities are for the recognition process. Therefore, we designed an experiment in which the variant probabilities were removed (thus replaced by 1) before compiling the variants of a word into one network, called the word pronunciation model. The first result was that the word pronunciation models had less states, yielding some extra efficiency in terms of CPU-time and memory requirements.

By supplying the 1021 rules mentioned in the previous section to the variant generation process, the variants were created as before, but now their probabilities were removed before compiling them into the word pronunciation models. The ASR performance was again measured as a function of the average number of variants per word (see Figure 3).

Fig. 2. WER/WER(BS) (in %) as a function of the number of variants/word. To control this number, \(P_{min}\) was changed from 0.03 to 0.25. The three panels correspond to values of 2, 1 and 0 for \(N_{cont}\).

For the closed vocabulary situation, the improvement with respect to the baseline system is 32% to 40%, which is just a little less than the 30...47% that was reported in [2]. However, the choice of \(P_{min}\) is now much more critical than before. The best performance is only reached in a narrow area. For the open vocabulary situation, on the other hand, the improvement is around 25%, which is larger than the 20% obtained with the full-scale rule learning process. This improvement also seems to remain quite stable as a function of the number of variants per word. The total performance (SX+SI) is about the same as before.

Comparing this figure to the first panel in Figure 2, it is clear that the improvement over the baseline system (single pronunciation per word) is reduced significantly for the closed vocabulary case: no more than 15% improvement anymore. Furthermore, the ASR performance degrades very rapidly with an increasing number of variants per word. With more than 3 variants per word, the system with variants performs even worse than the baseline system without variants.

This result confirms that the variant probabilities generated by the variant generation process are of critical im-
importance to the success of our method.

VIII. Conclusion

The first conclusion of this paper is that although this puts less emphasis on the negative rules, the introduction of an extra rule pruning step in the early phases of the rule learning process can speed up this process by a factor of 9 without causing any significant loss in the attainable ASR performance. The other two conclusions are that in order to benefit largely from pronunciation variation modeling in ASR, one has to take proper account of cross-word rules and variant probabilities.

References


