EFFICIENT RULE SCORING FOR IMPROVED GRAPHEME-BASED LEXICONS

William Hartmann, Lori Lamel, and Jean-Luc Gauvain

Spoken Language Processing Group, LIMSI-CNRS
91403 Orsay, France
{hartmann, lamel, gauvain}@limsi.fr

ABSTRACT

For many languages, an expert-defined phonetic lexicon may not exist. One popular alternative is the use of a grapheme-based lexicon. However, there may be a significant difference between the orthography and the pronunciation of the language. In our previous work, we proposed a statistical machine translation based approach to improving grapheme-based pronunciations. Without knowledge of true target pronunciations, a phrase table was created where each individual rule improved the likelihood of the training data when applied. The approach improved recognition accuracy, but required significant computational cost. In this work, we propose an improvement that increases the speed of the process by more than 80 times without decreasing recognition accuracy.

***Index Terms*—** automatic speech recognition, grapheme-based speech recognition, pronunciation learning

1. INTRODUCTION

Not all languages have the benefit of hand-crafted, highly accurate lexicons. Even hand-crafted lexicons may have inconsistencies, errors, or missing words. A common approach to handling a language without a predefined lexicon is through the use of graphemes. Graphemes provide a simple one-to-one mapping between words and a pronunciation—at least for the pronunciation dictionary used by the ASR system, though it does not necessarily correspond to the phonetics of the language. Depending on the relationship between the orthography and the phonetics of a given language, the graphemic representation can provide comparable performance to a phonetic-based lexicon [1]. Recent work has shown that the gap between phonetic and graphemic systems can be closed further by increasing the amount of training data [2].

Alternative or additional pronunciations for words can be learned directly from the data [3, 4]. Given a pre-existing acoustic model, pronunciation variants are generated by decoding the original training data. Several issues exist surrounding this approach. Only words that have been seen during training are affected; unseen words will keep their canonical pronunciations, possibly increasing the mismatch between the training and testing data. The new pronunciations can also increase the confusability of the lexicon by increasing the number of homonyms, significantly impacting recognition accuracy [5].

Many approaches exist to improve grapheme-based lexicons by converting the pronunciations to a different acoustic unit set, typically phones. Some of the first approaches used hand-derived rules to perform the conversion [6]. More recent approaches automatically learn a mapping from graphemes to phonemes [7] from a set of training data. This removes the necessity of using expert knowledge. In all cases the approaches are able to generalize to unseen words. While recent work has been done to reduce the amount of training data required [8], these approaches require at least some amount of training data to exist—training data that does not exist for our task.

An alternative to modifying the pronunciation lexicon is to capture the variations in a confusion model [9]. During decoding the confusion network expands the search space to implicitly allow for more pronunciation variation in the lexicon. This approach is best suited for capturing variation in an already well-defined lexicon. If the original pronunciation for a word is poor, it will still negatively affect the system. Also, the confusion model is only used during decoding; knowledge of potential confusions does not alter the training of the original acoustic models.

In this work, we focus on improving a pre-existing lexicon. Since no training data and no pre-defined target pronunciations exist, we require an alternative approach to transforming the pronunciations. In a prior study, we proposed a method for automatically discovering acoustic units and creating a pronunciation dictionary from an initial grapheme-based system [10]. The method for creating the pronunciation lexicon worked by transforming a baseline dictionary using a statistical machine translation (SMT)-based approach. The crucial component was the scoring of the individual rules used in the phrase table. Our proposed approach was computationally expensive and would be difficult to apply to larger datasets. In this work, we propose an improvement that allows the scoring to be performed more than 80 times faster than the previously proposed approach. Instead of scoring each rule individually, the entire phrase table is evaluated jointly.

In Section 2 we describe both the previous approach to rule scoring and the more efficient approach proposed in this work. Section 3 describes the the experimental setup. Results in terms of both word error rate and computational cost are presented in Section 4. Conclusions are presented in Section 5.

2. PRONUNCIATION TRANSFORMATION

Our general approach to pronunciation transformation is identical to the one proposed in [10], only the rule scoring method is altered. In other work, this is referred to as grapheme-to-phoneme (G2P) conversion [11, 12]. Since we are translating from graphemes to graphemes—or more generally, between two identical symbol sets—we refer to it as grapheme-to-grapheme (G2G) conversion. As in our previous work, we use a SMT-based approach. An SMT-based G2G system consists of a set of rules—referred to as a phrase table—that translate a sequence of symbols into an alternate sequence of symbols; in our study, the symbols represent acoustic units.

In order to create the phrase table, a set of training data is required. We build the training data by decoding the acoustic training data with context-independent (CI) acoustic units. Since the CI acoustic units have far fewer models than context-dependent (CD) units, we use 128 mixtures for each GMM. As noted in prior work, CD models produce more consistent pronunciations [13], but our goal at this stage is to produce as many reasonable pronunciation hypotheses as possible. This process generates a set of pronunciation hypotheses for each word in the lexicon.

Given the pronunciation hypotheses, the phrase table is created. If the phrase table was used to directly transform the original lexicon, it would significantly decrease the performance of the resulting ASR system. In many cases, a held out set of training data is used to further tune the parameters of the SMT system [14]. Unfortunately, true target pronunciations do not exist, so we cannot tune the parameters in this manner. In fact, we have experimented with approaches to tuning the SMT system such that the transformed lexicon minimizes the WER when using previously trained models. We have found that simply learning the optimal weights in this manner still does not improve overall performance. This is likely due to the amount of noise in the original training set. Our initial training step introduces a large number of rules that negatively affect performance. Tuning weights associated with the rule scores provided by Moses is not sufficient to reduce their effect.

Instead, we select the subset of rules in the phrase table that will result in an improved lexicon. The previously proposed procedure is described in Section 2.1, while the more efficient approach proposed in this work is described in Section 2.2. Once the individual rules have been scored, the original phrase table is pruned to contain only rules that surpass a certain threshold. The pruned phrase table is used to transform the original pronunciations into an improved lexicon.

2.1. Isolated Rule Scoring

This approach focuses on selecting rules that improve the overall likelihood of the training data. The procedure is outlined in Algorithm 1. Each rule in the phrase table is scored individually. Given the log-likelihood of the training data using the original lexicon, the average change to the log-likelihood is computed by applying each rule. This average change in log-likelihood becomes the score for the rule. Depending on the size of the training set, the number of utterances that need to be examined for each rule can be quite large. We artificially limit the number of utterances for each rule to 100 to reduce the computational cost.

Algorithm 1 Isolated Rule Scoring Procedure [10]

```
Input: set of training utterances $T$, unscored phrase table $P$,
  default lexicon $L$.
  For each rule $p_i \in P$.
      Let $L'$ be the transformed lexicon after applying rule $p$
      to $L$.
      Let $T'$ be the set of utterances in $T$ containing an altered word in $L'$.
      For each $t_j \in T'$
          $s_j$ = $s_j$ + log-likelihood change in $t_j$.
      $s_j$ = $s_j$ / size($T'$).
Output: set of scores $S$ for rules in $P$.
```

2.2. Single Pass Scoring

The previously described procedure is required to examine a subset of the training data for each rule in the phrase table. We propose a more efficient approach that only requires a single pass through the entire training set. The algorithm is presented in Algorithm 2. As opposed to focusing on the likelihood of the training data, we find the rules that are most frequently used when force aligning the data. A new lexicon is only computed once, containing all possible pronunciations of each word based on the rules in the phrase table. Each training utterance is aligned to find the best pronunciation for each word for that utterance. The score for each rule becomes the ratio of the number of times a rule produced a pronunciation used during forced alignment and the number of times a rule could have been selected. Another possible advantage of not scoring the rules independently is that the score is partially dependent on the interaction between rules.

While similar to simply selecting the most frequent pronunciation for each word, there are several important distinctions. Many words in the training lexicon are only seen a small number of times, with no single pronunciation being seen more frequently than any other—obviously the issue is even worse for words not seen during training. By instead focusing on frequent translations of graphemes, the approach can generalize to unseen words. Also, while one pronunciation of a word may not dominate, a portion of its pronunciation might.

Algorithm 2 Single Pass Scoring Procedure

```
Input: set of training utterances $T$, unscored phrase table $P$,
  default lexicon $L$.
  Let $A$ and $B$ be two count vectors of equal size to $P$.
  Let $L'$ be the lexicon containing all possible pronunciations
  for each word after applying $P$ to $L$.
  For each $t_j$ in $T$
      Let $D \subseteq L'$ contain all words in $t_j$.
      Let $W \subseteq D$ contain all pronunciations used in the
      forced alignment of $t_j$.
      If $p_i \in P$ was used to produce a pronunciation in $W$,
      increment $a_i \in A$.
      If $p_i \in P$ was used to produce a pronunciation in $D$,
      increment $b_i \in B$.
  For each $p_i \in P$, Let $s_i \in S = a_i / b_i$
Output: set of scores $S$ for rules in $P$.
```

3. EXPERIMENTAL SETUP

All speech recognition systems are built and tested using the HMM toolkit (HTK) [15]. The acoustic model uses cross-word triphones; each triphone has three states, modeled by a mixture of 16 Gaussians per state. Transition probabilities are tied across all models with the same center unit. Individual states are clustered across models, resulting in approximately 2000 tied states. State clustering is typically based on questions relating to phonetic classes. We do not assume this information is available for the acoustic units evaluated in this study. Instead, we use singleton questions (one question per acoustic unit) as is used in other work [1]. Decoding is performed with a bigram language model.

In addition to grapheme-based acoustic units, we also explore the use of automatically discovered acoustic units. We use the same unit discovery procedure as described in [10]. Three-state context-dependent grapheme models are clustered using spectral clustering [16] to generate the new acoustic units. Since each context dependent grapheme can be directly mapped to a single acoustic unit, an initial lexicon can be derived through this mapping.

To perform the pronunciation transformation, we use the Moses toolkit [17]. For each acoustic unit type, a phrase table is trained on approximately 70,000 word and hypothesized pronunciation pairs. The initial phrase table contains 500k rules, and it is too large to score using the previously proposed approach described in Section 2.1. We reduce the number of rules by pruning rules that were rarely seen in training, resulting in approximately 25k rules. This pruning is also done for the new approach for fair comparison. Once the rules are scored using one of the methods from Section 2, the phrase table is further pruned to only keep rules that pass a certain threshold. The original pronunciations are finally transformed by applying the reduced phrase table.

Evaluations are performed on the WSJ0 corpus, an English language 5000-word closed vocabulary task. The training set consists of 7,138 utterances from 83 speakers for a total of 14 hours of speech. The test set consists of 330 utterances from 8 speakers not seen during training. All settings were tuned using the development set. In this work, English was chosen because it allows for a comparison against using a hand-crafted dictionary and it has complex letter-to-sound rules [1].

4. RESULTS

WER results are presented in Table 1. A grapheme-based system performs significantly worse on this dataset compared to a similarly trained phone-based system (WER 8.0%). The first column contains the type of acoustic unit and the second column lists the number of acoustic units. The discovered units are the units built by clustering the original grapheme-based models, as described in [10]. Each result column describes the type of pronunciation dictionary used. Baseline pronunciations are the default lexicons without any type of transformation applied. *Isolated Rule* applies the SMT-based pronunciation transformation procedure using a phrase table scored by the procedure described in Section 2.1. *Single Pass* uses the the more efficient scoring procedure proposed in Section 2.2. We also experimented with larger numbers of acoustic units, but it did not produce further gains.

The *Single Pass* scoring procedure produces similar results to *Isolated Rule*, but is approximately 80 times faster. While the WER improvements produced by *Single Pass* over *Isolated Rule* are not statistically significant, the improvement between the baseline grapheme-based system and the best performing transformed system is increased from 13% relative WER to 16%. In addition, the difference between the baseline lexicon and the *Single Pass* lexicon in each row are statistically significant ($p \leq 0.05$).

The main contribution of *Single Pass* over *Isolated Rule* is the dramatic reduction in the time required to score the rules. Figure 1 compares the running time of the two methods on a single 2.0 GHz processor. Note that the y-axis—computation time in minutes—is a logarithmic scale. The difference between the two scoring methods is so large that it would be difficult to see on a linear scale. *Isolated Rule* takes approximately 30 seconds per rule and the running time is linear in the number of rules. *Single Pass* has a significantly reduced running time, growing very slowly in terms of the number of rules. The increased cost for each additional rule is negligible. The difference in time between the two methods when scoring the full phrase table is nearly two orders of magnitude.

Unit Type	# Acoustic Units	Baseline	Isolated Rule	Single Pass
Grapheme	26	15.8	14.5	14.2
Discovered	39	15.0	13.9	13.3
Discovered	50	15.2	13.9	14.1
Discovered	60	**14.4**	**13.8**	**13.2**

Table 1. Results for both grapheme-based acoustic units and automatically discovered acoustic units (Section 3) in terms of WER (%). Baseline are the original pronunciations while the final two columns use the transformed lexicons. Isolated Rule is the previously proposed approach described in Section 2.1 and Single Pass is the improved approach described in Section 2.2.

Note that both approaches are easily parallelizable.

Single Pass has a minimum computational cost, the time required to force align the training set, regardless of the number of rules in the phrase table. As the size of the phrase table increases, the time to generate all possible pronunciations also increases. Since this computation is only performed once for the entire dataset, the overall cost is small. A secondary effect is the number of pronunciations for each word increases, slightly increasing the time required to force align each sentence. However, the majority of the additional cost comes from determining the rules associated with each pronunciation. By precomputing these associations, especially for frequent words, the computational cost could be further reduced.

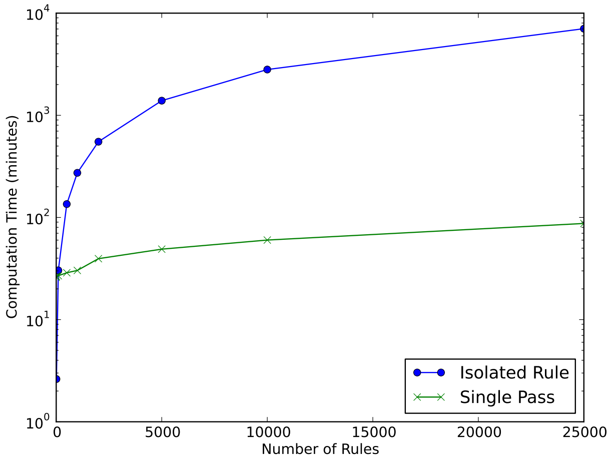

Fig. 1. Number of minutes required for computation versus the number of rules scored for the two methods described in Section 2. Note that the y-axis is on a logarithmic scale. *Single Pass* is nearly two orders of magnitude faster than *Isolated Rule*.

5. CONCLUSIONS

We have presented a method for improving a grapheme-based lexicon by transforming the original pronunciations using a SMT-based approach. Since target pronunciations are not known, the approach relies on pruning the rules in the phrase table as opposed to the typical approach of tuning weights. The general approach was previously proposed in [10], but due to the computational cost, was unlikely to scale to larger datasets. In this work, we presented an alternative scoring method that is more than 80 times faster than the previously proposed method, while obtaining a small improvement in recognition performance. The relative WER improvement with the new method is 16% (compared to 13% with the previous method). With the reduced computational cost, it is now feasible to perform experiments on larger datasets. We have begun experiments on several under-resourced languages (Tagalog, Turkish, and Pashto) from the IARPA Babel project. We will explore how the results are affected by the amount of correlation between the orthography and phonetics of a language.

6. ACKNOWLEDGEMENTS

This research was partially supported by OSEO, the French State agency for innovation, under the Quaero program and by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense US Army Research Laboratory contract number W911NF-12-C-0013. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U.S. Government.

REFERENCES

[1] M. Killer, "Grapheme-based speech recognition," M.S. thesis, Carnegie Mellon University, 2003.

[2] Y.-H. Sung, T. Hughes, F. Beaufays, and B. Strope, "Revisiting graphemes with increasing amounts of data," in *Proceedings of IEEE ICASSP*, 2009, pp. 4449–4452.

[3] M. Weintraub, E. Fosler, C. Galles, Y.-H. Kao, S. Khudanpur, M. Saraclar, and S. Wegmann, "Ws96 project report: Automatic learning of word pronunciation from data," in *JHU Workshop Pronunciation Group*, 1996.

[4] I. Badr, I. McGraw, and J. Glass, “Learning new word pronunciations from data,” in *Proceedings of Interspeech*, 2010.

[5] W. Hartmann and E. Fosler-Lussier, “Investigating phonetic information reduction and lexical confusability,” in *Proceedings of Interspeech*, Brighton, England, 2009, pp. 1659–1662.

[6] R. Kaplan and M. Kay, “Regular models of phonological rule systems,” *Computational Linguistics*, vol. 20, no. 3, pp. 331–378, 1994.

[7] M. Bisani and H. Ney, “Joint-sequence models for grapheme-to-phoneme conversion,” *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[8] N. Goel, S. Thomas, M. Agarwal, P. Akyazi, L. Burget, K. Feng, A. Ghoshal, O. Glembek, M. Karafiat, D. Povey, A. Rastrow, R. C. Rose, and P. Schwarz, “Approaches to automatic lexicon learning with limited training examples,” in *Proceedings of IEEE ICASSP*, 2010, pp. 5094–5097.

[9] P. Karanasou, F. Yvon, T. Lavergne, and L. Lamel, “Discriminative training of a phoneme confusion model for a dynamic lexicon in asr,” in *Proceedings of Interspeech*, 2013.

[10] W. Hartmann, A. Roy, L. Lamel, and J. L. Gauvain, “Acoustic unit discovery and pronunciation generation from a grapheme-based lexicon,” in *Proceedings of IEEE ASRU*, 2013, pp. 350–355.

[11] Panagiota Karanasou and Lori Lamel, “Pronunciation variant generation using SMT-inspired approaches,” in *Proceedings of IEEE ICASSP*, 2011, pp. 4908–4911.

[12] A. Laurent, P. Deléglise, and S. Meignier, “Grapheme to phoneme conversion using an SMT system,” in *Proceedings of Interspeech*, 2009, pp. 708–711.

[13] M. Adda-Decker and L. Lamel, “Pronunciation variants across systems languages and speaking style,” in *Proceedings of ESCA Workshop on Modeling Pronunciation Variation for Automatic Speech Recognition*, 1998, pp. 1–6.

[14] F. J. Och, “Minimum error rate training in statistical machine translation,” in *Proceedings of the Association for Computational Linguistics*, 2003, vol. 1, pp. 160–167.

[15] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*, Cambridge University Publishing Department, 2002.

[16] A. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” *Advances in Neural Information Processing*, vol. 14, pp. 849–856, 2002.

[17] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, “Moses: Open source toolkit for statistical machine translation,” in *Proceedings of the Association for Computational Linguistics*, 2007, pp. 177–180.